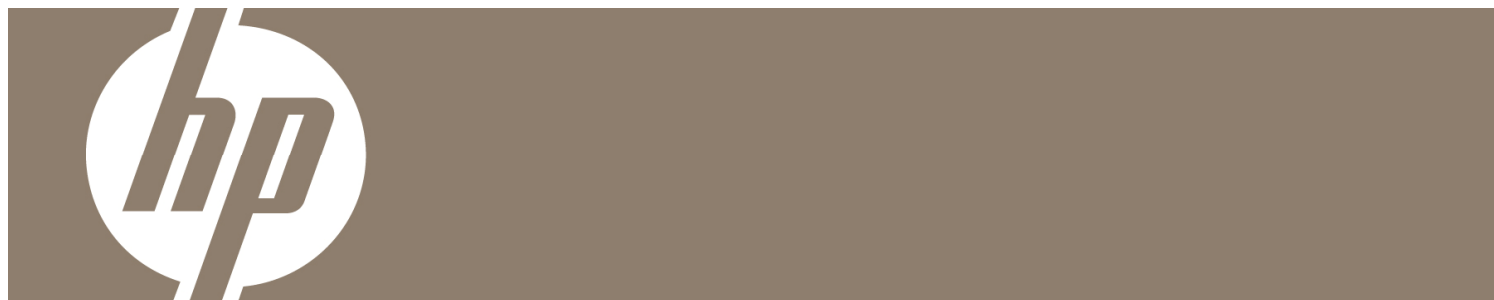


Best Practices for Using Integrity Virtual Machines



Introduction	3
Basics – Read This First	3
Take Advantage of HP-UX on the Host.....	4
System Management and Security.....	4
Workload Management.....	4
Instant Capacity.....	4
Monitoring.....	5
Virtual Machine Definition.....	6
Create Virtual Machines Conservatively	6
Uniprocessor Virtual Machines are More Efficient than Virtual SMPs.....	6
Use Physical Memory Wisely.....	6
Getting the Most from Virtual I/O.....	7
Virtual Mass Storage	7
General Flexibility	9
Flexibility vs. Performance	10
Storage Management and Fault Tolerance.....	11
Using Multi-Path Technologies with Integrity VM.....	12
Managing Virtual Storage on the VM Host.....	12
Stripe Disks On the Host Where Possible	13
Summary.....	13
Virtual DVD.....	13
Performance	13
Ease of Use	13

Getting the Most From Virtual DVD	14
Virtual Networking	15
Reliability and Performance	15
Adding Network Capacity	15
Leverage Existing Topologies	16
Tuning Virtual Machines for Applications	18
For more information	19

Introduction

As is the case with any relatively new technology, there are a lot of questions surrounding basic use of that technology, tips on how to get the most out of that technology, and common pitfalls for the new user. This white paper provides an assortment of guidelines for deploying one or more virtual machines (VMs) with the Integrity Virtual Machines product. The scope of this white paper is limited to the Integrity Virtual Machines product, also referred to simply as Integrity VM. Several other HP products are discussed and references for more information on those products are provided in the appropriately-named 'For more information' section of this paper.

Basics – Read This First

In the event you are about to do your first deployment or are troubleshooting problems with an existing deployment, take a moment to make sure you have a supported configuration. Common problems are often addressed by checking a few basic items, including:

- The physical Integrity Server with the Integrity Virtual Machines product installed on it is commonly referred to as the Integrity VM Host or simply the VM Host. The VM Host must be running a version of HP-UX supported by Integrity VM. Version 1.2 and 2.0 of Integrity VM are supported on HP-UX 11iv2 0505, 0512, 0603, and 0606. Version 2.0 is also supported on 11iv2 0609.
- The Integrity VM Host may require a few patches, depending on the version of HP-UX it is running. These will be listed in the Integrity VM release notes along with instructions for obtaining and installing them.
- Make sure the OS running on the virtual machine, i.e., the guest, has the correct patches and tunable settings for the workload running there. Do not modify the tunable settings on the VM Host to accommodate a workload executing in a virtual machine. Do not modify the VM Host's tunables as they will be modified during installation of Integrity VM to provide optimal performance and functionality for hosting virtual machines.
- Read the Integrity VM release notes.

Take Advantage of HP-UX on the Host

The Integrity VM software includes the HP-UX 11i Foundation Operating Environment (OE) which executes on the Integrity Server used to host virtual machines. Having the HP-UX 11i Foundation OE on the VM Host provides all of the basic HP-UX functionality including HP system management tools as well as a typical UNIX environment. Moreover, the VM Host supports the same devices and peripherals as any other Integrity Server running HP-UX 11i. Having the Foundation OE on the VM Host leverages the extensive testing, reliability and maturity of the HP-UX 11i operating system.

System Management and Security

Device and overall system management of the VM Host benefits from the availability of existing management tools such as System Management Homepage (SMH). Configuration and management of storage devices, a critical task in deploying virtual machines, is made easy with these tools.

Both HP's Logical Volume Manager (LVM) and the VERITAS Volume Manager (VxVM) may be used for storage configuration and management on the VM Host. That is, logical volumes created on the VM Host with either LVM or VxVM can be used as virtual storage for VMs. Integrity VM version 2.0 and later is compatible with versions 3.5 and 4.1 of VxVM.

Software deployment and management on the VM Host is enhanced by the availability of

- Ignite-UX
- Software Distributor-UX
- Update-UX
- Patch Assessment Tool

Security of the VM Host and the related benefits to the virtual machines running there is extended by the availability of both HP-UX Bastille and the Security Patch Check tools.

Workload Management

The workload management tool of choice for Integrity Virtual Machines is HP's global Workload Manager (gWLM). It is closely integrated with the Integrity VM product and provides a rich set of features for managing virtual machines as workloads. Many other workload management products are not supported for use on the VM Host itself, including HP's Process Resource Manager (PRM). The Workload Manager (WLM) product may be used on the VM Host to manage that system's instant capacity resources, but it cannot be used to manage the virtual machines. Partitioning technologies such as virtual partitions (vPars) and processor sets (PSETs) are not supported on the VM Host.

The GlancePlus Pak may also be used for virtual machine workload management, providing all the familiar concepts and use cases common to the Glance and OpenView Performance Agent products as well as integration with OpenView management products.

Instant Capacity

HP's high-end Integrity Servers provide immediate access to additional capacity with the Instant Capacity and Temporary Instant Capacity products. Having the option of instant availability to additional resources on the VM Host is a powerful tool in addressing increases – permanent or temporary – in aggregate resource demand from virtual machines running there. For example, when Instant Capacity on the VM Host is used to enable additional CPU resources they are immediately available to the Integrity VM scheduler for allocation to the virtual machines running there.

Monitoring

Since each VM is manifested as a UNIX process running on the VM Host, the physical resources – including CPU, I/O, etc. – consumed by a given VM can be identified by monitoring the process associated with that VM. These processes have the executable name `hpvmappp` and typically have the option `-d` whose argument name is the name of the VM. For example, the process with command `'hpvmappp -d vm01'` corresponds to the virtual machine named 'vm01.'

Simple tools such as `ps` and `top` can be used on the VM Host to monitor a virtual machine by identifying the process ID for a given VM. For example, the PID for some VM can be identified from the output of `'ps -fu root | grep hpvmappp'` and then used with `top` to identify the resources being consumed by that VM.

More elegant solutions can be achieved with tools such as HP's GlancePlus performance monitoring tool. Each VM may be defined as an OpenView application by creating an application definition in the OpenView parameter file `/var/opt/perf/parm`. For example, inserting the following application definitions for the virtual machines named `vm01`, `vm02`, and `vm03` in `/var/opt/perf/parm` enables GlancePlus to identify them as applications:

```
application vm01
cmd = *hpvmappp -d*vm01

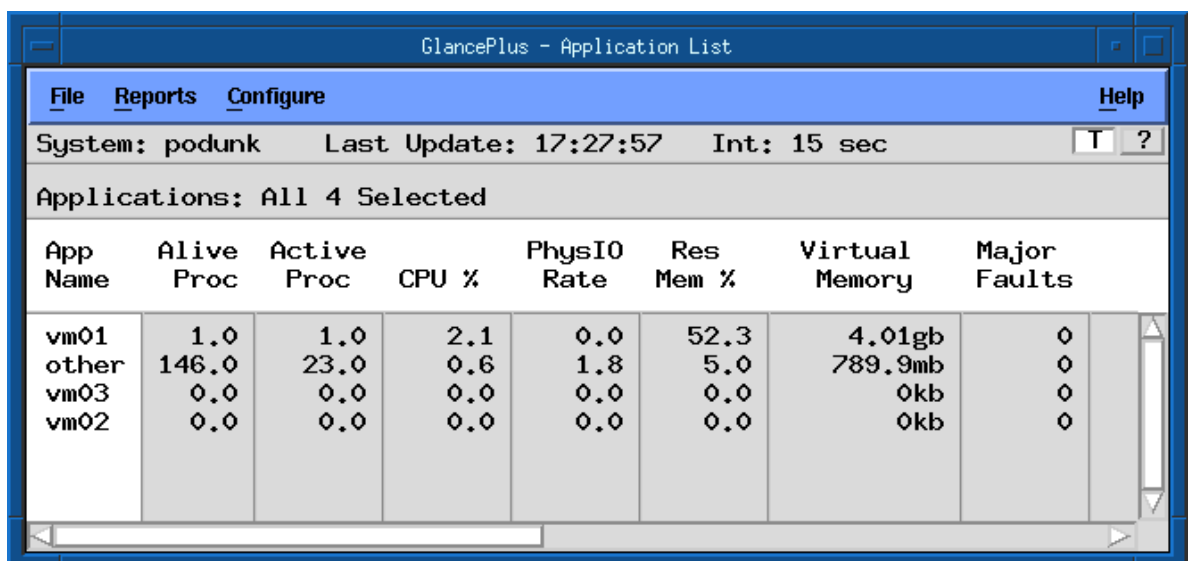
application vm02
cmd = *hpvmappp -d*vm02

application vm03
cmd = *hpvmappp -d*vm03
```

This makes it easy to track the resource utilization of each VM through Glance's Application List reporting functionality.

Figure 1 provides an example of GlancePlus visualization of VMs using the application definitions mentioned above.

Figure 1 - GlancePlus can be used to monitor VMs as Applications



The screenshot shows a window titled "GlancePlus - Application List" with a menu bar (File, Reports, Configure, Help) and a status bar (System: podunk, Last Update: 17:27:57, Int: 15 sec). Below the status bar, it says "Applications: All 4 Selected". The main content is a table with the following columns: App Name, Alive Proc, Active Proc, CPU %, PhysIO Rate, Res Mem %, Virtual Memory, and Major Faults. The data rows are:

App Name	Alive Proc	Active Proc	CPU %	PhysIO Rate	Res Mem %	Virtual Memory	Major Faults
vm01	1.0	1.0	2.1	0.0	52.3	4.01gb	0
other	146.0	23.0	0.6	1.8	5.0	789.9mb	0
vm03	0.0	0.0	0.0	0.0	0.0	0kb	0
vm02	0.0	0.0	0.0	0.0	0.0	0kb	0

Virtual Machine Definition

Create Virtual Machines Conservatively

Once you begin to define a virtual machine, it is very tempting to create it as large as possible – it's all virtual hardware, right? While it is true that a VM uses virtual hardware, there are some repercussions to creating it with more virtual hardware than it needs.

Uniprocessor Virtual Machines are More Efficient than Virtual SMPs

Generally speaking, uni-processor systems are more efficient than multi-processor systems. Scheduling, memory access, and resource contention issues are all easier if only one processor is involved. The same is true for virtual machines.

Moreover, each virtual processor is allocated some minimum fraction of a physical processor's resources. So a VM with four virtual processors requires four times the processor resources that a VM with a single processor requires. Each of virtual processor from a virtual SMP must be allocated resources from separate physical processors – two virtual processors from the same virtual machine cannot be scheduled on the same physical processor. The CPU resources of a VM Host may be prematurely exhausted by the definition of a virtual SMP.

For example, suppose the VM Host has four physical processors and a VM with 4 virtual processors, each with an entitlement of 25%. This would leave one to believe that 3 uni-processor VMs with 100% CPU entitlement can also be created since the remaining CPU resources on the VM Host is a total of 300%. Not true. The resource guarantee mechanism in Integrity VM requires that 25% of four physical CPUs be available for the original virtual SMP (with four virtual CPUs). That leaves only 75% of three physical CPUs available for virtual processors. A virtual CPU may be scheduled on exactly one physical CPU, so the largest entitlement any virtual CPU can receive is 75%. So, the largest entitlement those 3 uni-processor VMs can be defined with is 75%.

Use Physical Memory Wisely

The more memory any system is configured with, the more likely it is to incur page faults and other memory management interrupts. The same applies to virtual machines. Moreover, the memory used by a VM cannot be made available to other running VMs on that same VM Host. For both of these reasons, allocate memory to VMs conservatively.

Getting the Most from Virtual I/O

Virtual Mass Storage

Given the availability of so many storage technologies and means of securing them, the versatility of Integrity VM mass storage provides a multitude of options. However, there are scenarios where some of these options are not prudent.

In general, the best I/O performance for VMs is obtained by mapping virtual disks directly to entire physical disks (or LUNs corresponding to SAN storage). Workloads that require large storage capacity should define storage in this way wherever possible. This approach is illustrated in Figure 2. Doing so will provide the workloads with access to all the performance capability of these disks, not having to share them with other workloads. In such scenarios, the only reason to use files or logical volumes for virtual hard disks is convenience. Otherwise, the configuration may be unnecessarily complicated as outlined in

Figure 3.

Figure 2 – Whenever virtual mass storage capacity requirements are large, mapping virtual disks to physical storage is more efficient and less complex.

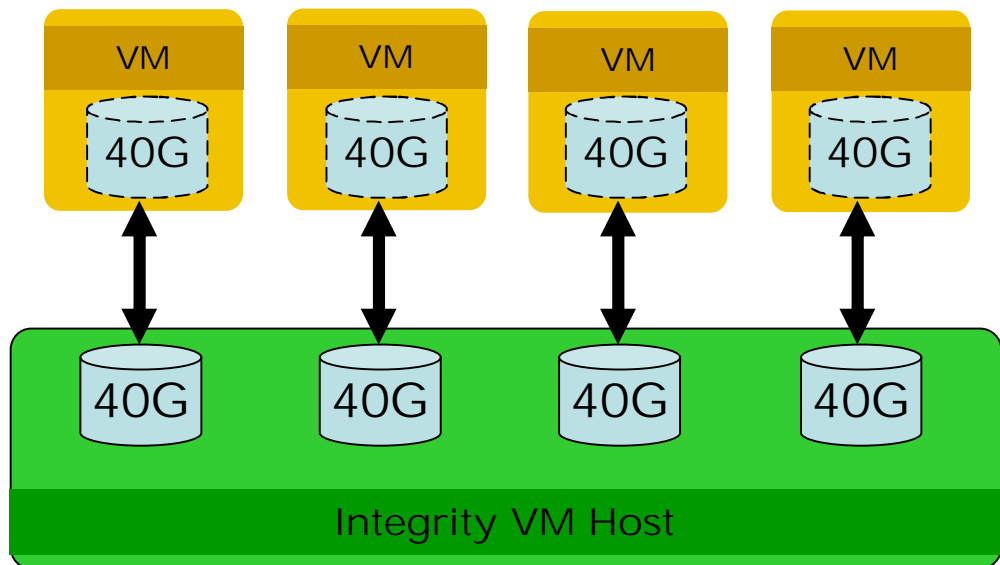
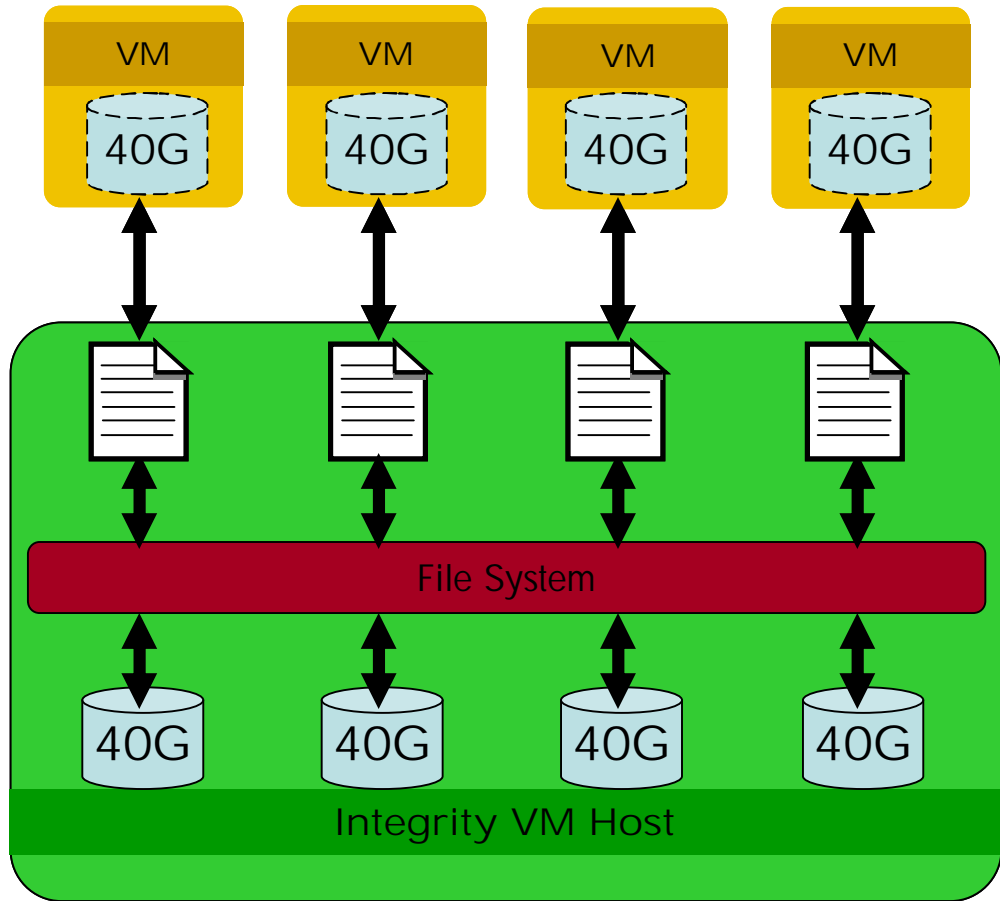


Figure 3 - Mapping large capacity virtual mass storage to files adds complexity and is not recommended unless it simplifies system management and administration.



General Flexibility

When considering which type of storage to use – file, logical volume, disk, or partition – on the VM Host, flexibility is an issue that may be of great importance in your VM configuration.

Logical volumes, created with VxVM or HP's LVM, are easy to extend, import, or export. Each of these actions may be applied on the VM Host to logical volumes serving as virtual disks – provided the VM using those logical volumes has been shut down and powered off.

Virtual disks in the form of files on the VM Host are the easiest to transport. They may be moved from one VM Host system to another in the same way any other file may be moved, for example, using ftp or rcp. This flexibility makes files an ideal candidate for use as virtual DVDs.

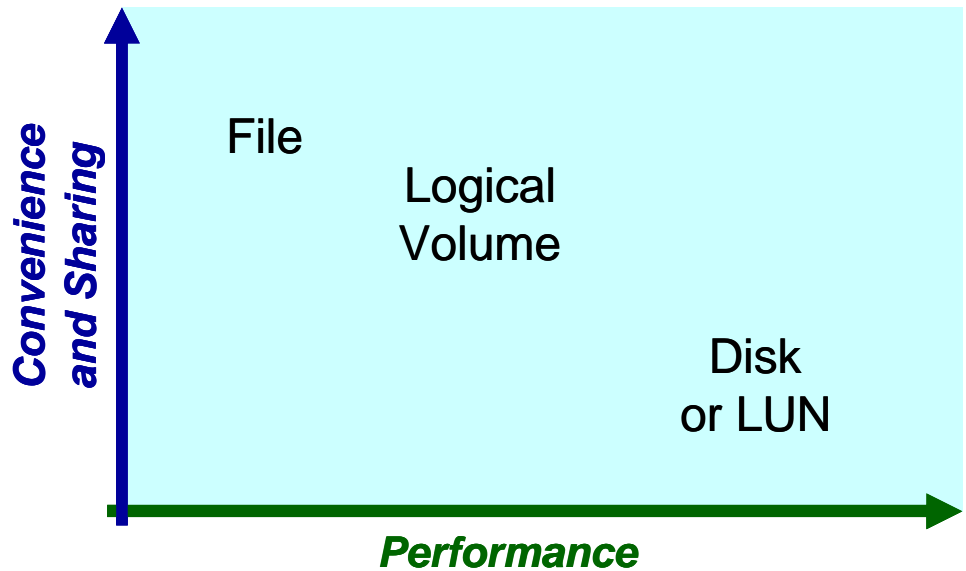
Moving physical disks is possible, but not convenient. Hence, the use of entire disks local to the VM Host limits flexibility. On the other hand, using LUNs associated with a SAN is arguably the most flexible option. If an entire LUN on a SAN is used by a VM, then it may be used by a VM on another VM Host provided that host has connectivity to that SAN storage. In fact, this means of storage is recommended for any VM that may be migrated (using `hpxmmigrate` for example) from one VM Host to another.

Finally, partitions offer the least flexibility in that they may not be modified in any way without losing data.

Flexibility vs. Performance

The various mass storage options for Virtual Machines each have a unique set of benefits. Virtual hard disks using files on the VM Host provides flexibility in sharing physical storage and the convenience of managing files on a UNIX system. Mapping a virtual hard drive directly to an entire physical disk (or LUN) provides the best performance, eliminating the overhead of moving I/O transactions through the volume management subsystem and/or file system on the VM Host. Logical volumes, as virtual hard disks, do offer better sharing than physical disks and better performance than files. Partitions (created with `idisk`) perform almost as well as disks, but are difficult to use. Figure 4 illustrates the general tradeoffs between the various storage options.

Figure 4 – Defining virtual hard disks using files, logical volumes, or physical disks present tradeoffs in convenience and performance



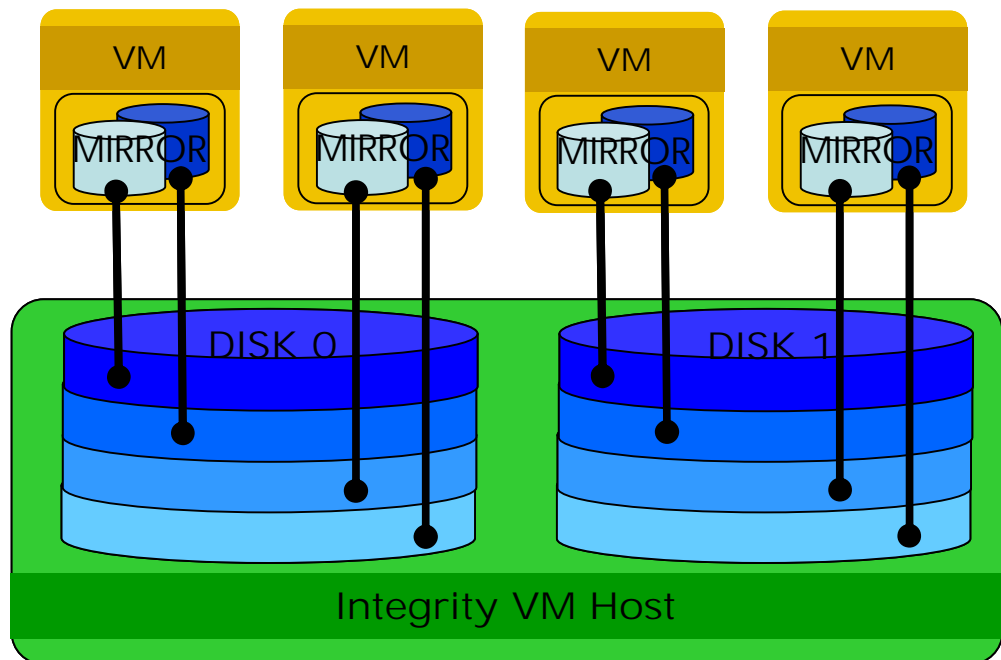
Storage Management and Fault Tolerance

Arguably one of the most powerful technologies in Integrity VM is its capability to use multiple entities – files, logical volumes, disks, etc. – as virtual mass storage (virtual hard drives) for VMs. As mentioned earlier, standard HP-UX tools can be used to manage storage on the VM Host. The same is true for fault tolerant strategies and solutions. RAID strategies, such as mirroring, should be done on the host. There are several reasons for this, including:

- Protecting the physical storage on the host automatically protects it for the VMs using that storage.
- Storage fault tolerance solutions need only be implemented once on the Host as opposed to implementing them multiple times – once for each VM.
- Storage fault tolerance on the VM will often be a waste of time because it protects against hardware failures which can not occur in virtual devices but only in physical devices.

To illustrate these points, consider the simple VM configuration in Figure 5. In this configuration, two physical disks on the VM Host are partitioned into four logical volumes each, for a total of eight logical volumes. Each of these is then used as virtual hard disks used by VMs – four VMs, each using two logical volumes. On each VM, the two virtual hard disks are mirrored for data protection. The problem with this approach is that if one of the physical disks on the VM Host should actually fail, then the mirroring on two of the VMs provided no benefit whatsoever – all four logical volumes would fail along with that physical disk.

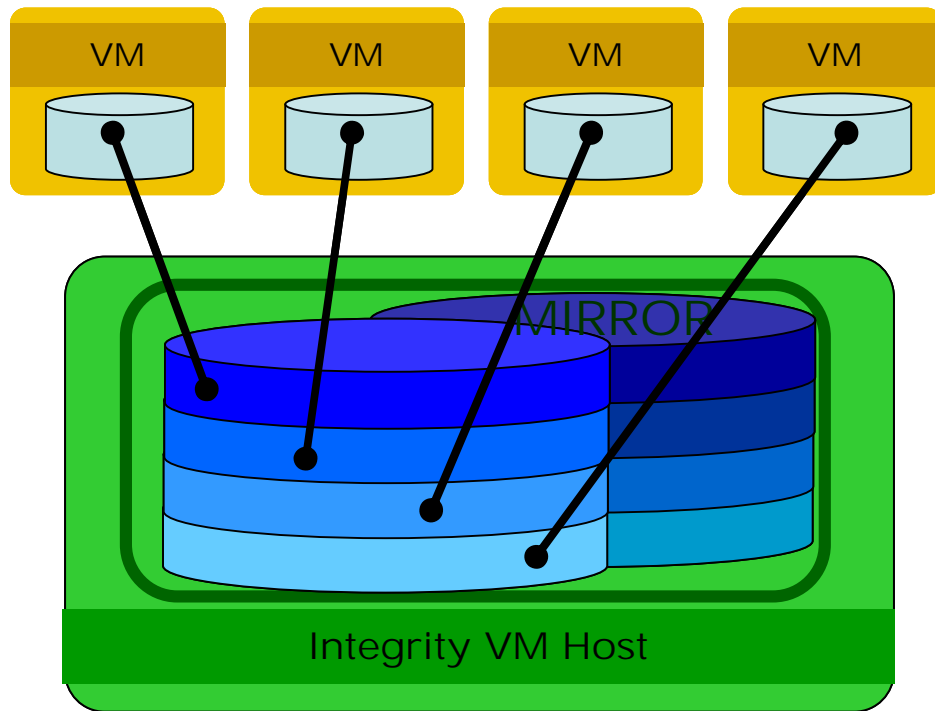
Figure 5 – Data protection in a VM usually does not provide true fault tolerance for storage – this is not a recommended configuration



Contrast this scenario with that in

Figure 6 where data protection is implemented on the VM Host. With this approach, the physical storage is protected and it need only be done once – on the VM Host. Similarly, mass storage arrays (with their own data protection implementations) connected to the VM Host should be used in this same way for the same reasons.

Figure 6 – Data protection on the VM Host protects virtual mass storage and is easier than repeating it on each VM



Using Multi-Path Technologies with Integrity VM

Logical volumes as virtual disks provide high availability since they have multi-path and mirroring capability built in. You also get a choice between LVM (PVLlinks) or VxVM (DMP) which allows further flexibility.

However, the VM Host has only one multi-path option for disks – Secure Path. Note that Secure Path has restrictions on which hardware devices it supports.

Files, being at the top of the IO stack, enjoy the ability to use any multi-path/mirroring/RAID option available on HP-UX. Also, since they are easy to move, they can be backed up on another system easily. However, if one doesn't set up HA for them using something underneath, then they have nothing to protect them from being destroyed.

Partitions can use 2 multi-path options in 11.23 – EMC PowerPath and Secure Path – however, they have to rely on the disk technologies for RAID functionality and they are not supported for use with off-line migration of VMs.

Any redundancy strategy should be implemented on the VM Host – replication on the VM is not supported. This applies to technologies such as SecurePath, PowerPath, and PVLlinks. The primary path to any storage unit should be the only path used by a VM as a virtual hard drive. That is, secondary paths must not be used by VMs as virtual hard drives.

Managing Virtual Storage on the VM Host

Disks are arguably the easiest logical storage to manage on the VM Host because the mapping of the virtual storage to physical storage is the simplest. This is a big advantage when one wants to make a change in the data center and understand what system(s) are impacted. Disk use can be identified with tools such as System Management Homepage (SMH) and `hprvmddevmgmt`. Disks are also the

easiest to set up because they are typically already set up on the VM Host and they typically only require `insf` to create a device file for them to be used as virtual storage for a VM

Logical volumes are fairly easy to manage with SMH and other tools. Administrators accustomed to setting up separate logical volumes are already familiar with such tools. The logical volume should be used in raw format – creating a file system there is a waste of time. A single physical disk may be more efficiently used by dividing it into multiple logical volumes, but this comes at a manageability price in the event that disk should fail.

File and file system management is familiar to all system users, making files the most convenient of all the virtual storage options. Integrity VM provides functionality with the `hpvmdevmgmt` command to create files for use as virtual storage. Clearly, naming of directories, files, etc is useful and important in understanding and organizing what VM is using which file for storage. While file system commands are easy to use, a single typographical error can destroy a lot of data. Remember to create the file systems such that they are capable of containing large files (larger than 2 GB).

Partitions are the least manageable simply because there are few tools on HP-UX to manage them. The `idisk` utility is difficult to use and its documentation misleads the user into believing that the partition type may be important for use as virtual storage. Any type will suffice, but one is forced to create an EFI partition first. This is the only one you will need for virtual storage, so make it sufficiently large for your planned utilization.

Stripe Disks On the Host Where Possible

Creating volume groups on multiple disks should be done on the VM Host wherever possible. This is especially true when the volume group is configured with striping across those disks. Striping across multiple disks on the VM Host delivers significantly better performance than virtualizing each of those disks and striping across them on the VM. This also minimizes the number of devices required on the VM.

Summary

Generally speaking, there are two rules of thumb when consolidating with VMs and configuring virtual devices for those VMs – keep it simple and, when in doubt do it on the VM Host.

Wherever possible, keep configurations simple. As mentioned earlier, if large mass storage is required then use whole physical disks for virtual disks – don't bother using logical volumes or files for them. The more complex the configuration, the more likely it is to become problematic

Virtual DVD

Virtual DVDs may be mapped to files or physical DVD drives (logical volumes cannot be used for virtual DVD).

Performance

One clear advantage of files over the use of physical DVDs is performance – file I/O will be faster than from a physical DVD. Moreover, such files may be designated as shared (using `hpvmdevmgmt`), allowing these virtual DVDs to be used by multiple virtual DVD drives in multiple VMs – simultaneously. Physical DVDs should not be used simultaneously by VMs and, hence, should not be designated as shared.

Ease of Use

Reflecting on the previous section, Virtual Mass Storage, it follows that files are easier to manage than physical DVDs. Once the media is available as a file, for example, as an ISO image, it need not be present in the physical DVD drive to be used by a virtual DVD. It is often a good practice to define the virtual DVD with bus/device/target of 0/0/0 to accommodate assumptions of where DVD is located (e.g., installing ISV software). This can be achieved by creating your VM such that its first resource (amongst disk, DVD, or network) is the virtual DVD drive and/or creating the virtual DVD

with explicitly bus, device, and target arguments (see the `hpvmcreate` and `hpvmmodify` commands for more details).

Getting the Most From Virtual DVD

Mapping virtual DVD drives to files is a powerful tool in deploying software on virtual machines. This is especially true for software that requires multiple disks for installation. To illustrate efficient use of virtual DVD functionality, consider the following example.

Suppose that a VM Host will be hosting multiple virtual machines running Windows Server. In addition to the Windows Server 2003 installation media from Microsoft, HP provides – as physical media or by download from the HP support web site – its Smart Setup media as well as patches and support provided in its Smart Update media. It can be tedious walking to and from the datacenter where the physical Integrity VM Host system is located to load and unload disks for such a software installation. To alleviate this repeated inconvenience, proceed as follows:

1. First, find sufficient file system space on the VM Host for copies of the physical installation disks. Copy the disks onto the file system using a utility such as `dd`. For example:

```
dd if=/dev/rdisk/c0t0d0 of=/hpvm/DVD/WinSvr2003.iso
```

Do this for each installation disk.

2. Create (or modify the existing) virtual DVD so that it maps to the first disk you need to install. For example, you may modify the virtual DVD without rebooting the VM as follows:

```
hpvmmodify -P myvm -a dvd:scsi:0,0,0:file:/hpvm/DVD/WinSvr2003.iso
```

3. Inside the VM, install the software as you normally would until finished with that disk.
4. Virtually remove the current disk and insert the next one. This is best done using the `hpvmmodify` command. For example:

```
hpvmmodify -P myvm -m dvd:scsi:0,0,0:file:/hpvm/DVD/SmartSetup.iso
```

You need not reboot the VM for this to be accomplished.

5. Inside the VM, continue the installation just as though you had physically ejected the old disk and inserted the new one.

Subsequent installations may now be done by repeating steps 2 through 5 above – without having to physically insert and eject the media. If a given file is to be used by multiple VMs for a virtual DVD, then you will need to modify that file's `SHARE` attribute in the HPVM device management database. For example, to identify the file `/hpvm/DVD/WinSvr2003.iso` as a shared device, use the `hpvmdevmgmt` command as follows:

```
hpvmdevmgmt -m gdev:/hpvm/DVD/WinSvr2003.iso:attr:SHARE=YES
```

Virtual Networking

Reliability and Performance

The HP Auto Port Aggregation (APA) product may be used on the VM Host to provide high availability for virtual networking. This is accomplished by defining your virtual switch(es) in terms of the port created by APA (for example, lan900) rather than the LAN numbers associated with the physical network interface cards (NICs). In doing so, all of the virtual NICs defined in terms of that virtual switch automatically benefit from the high availability and aggregate throughput provided by APA.

Adding Network Capacity

Network capacity can be increased in two basic ways – through the use of APA or definition of additional, unused, virtual NICs in a VM configuration.

A physical NIC may be added to a VM Host and an APA configuration without rebooting the VM Host. Due to the flexibility of APA, this card's connection is immediately available to the APA port and, hence, to the virtual switch using that APA port. The end result is an online increase in network bandwidth for all VMs using that virtual switch.

Alternatively, one may define multiple virtual NICs in a VM configuration without using them all. In particular, some 'spare' virtual NICs may be defined in terms of a virtual switch that does not exist. This will not prevent the VM from being started, nor will it prevent the installation of the operating system on that VM. These 'spare' virtual NICs will operate in a manner similar to any physical NIC without a network connection. When there is a need for additional network bandwidth, a physical NIC may be added to the VM Host (without rebooting) and then the virtual switch associated with these 'spares' can be created such that it is associated with this new physical NIC. Once the virtual switch is powered on, all of the virtual NICs will behave as if they had just been connected to the network (which, in fact, they have). Subsequently these virtual NICs may be configured and used by the VMs in the desired manner.

Leverage Existing Topologies

The topology of existing configurations is, typically, created for good reason. This may be more important for networking than for other devices. For example, if existing servers have two network connections – one for general use and another for backups then take care when virtualizing those connections as illustrated in Figure 7. Carelessness in configuring them can result in serious performance (and perhaps functionality) problems that did not previously exist. In this example, interactive response times may increase on one VM due to its sharing a physical NIC with multiple, traffic-intensive, backup connections.

Contrast this with Figure 8 which reflects the topology in the configuration of the virtual NICs so that they are associated with virtual switches connected to the appropriate network connections for general use and backups.

Figure 7 – Do not be careless in mapping virtual NICs to virtual switches as illustrated in this example. Performance and functionality problems can be created by not maintaining existing network topologies.

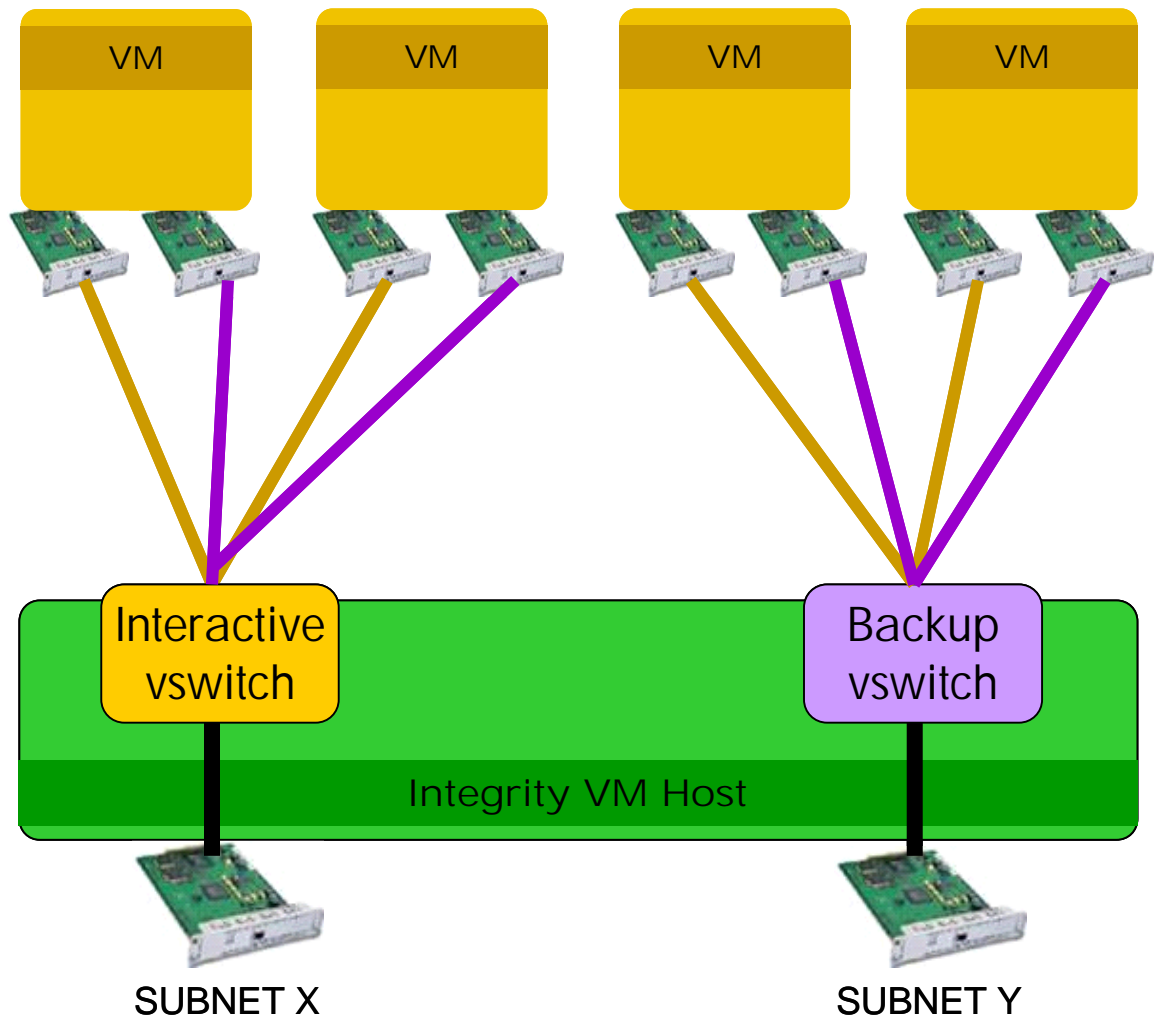
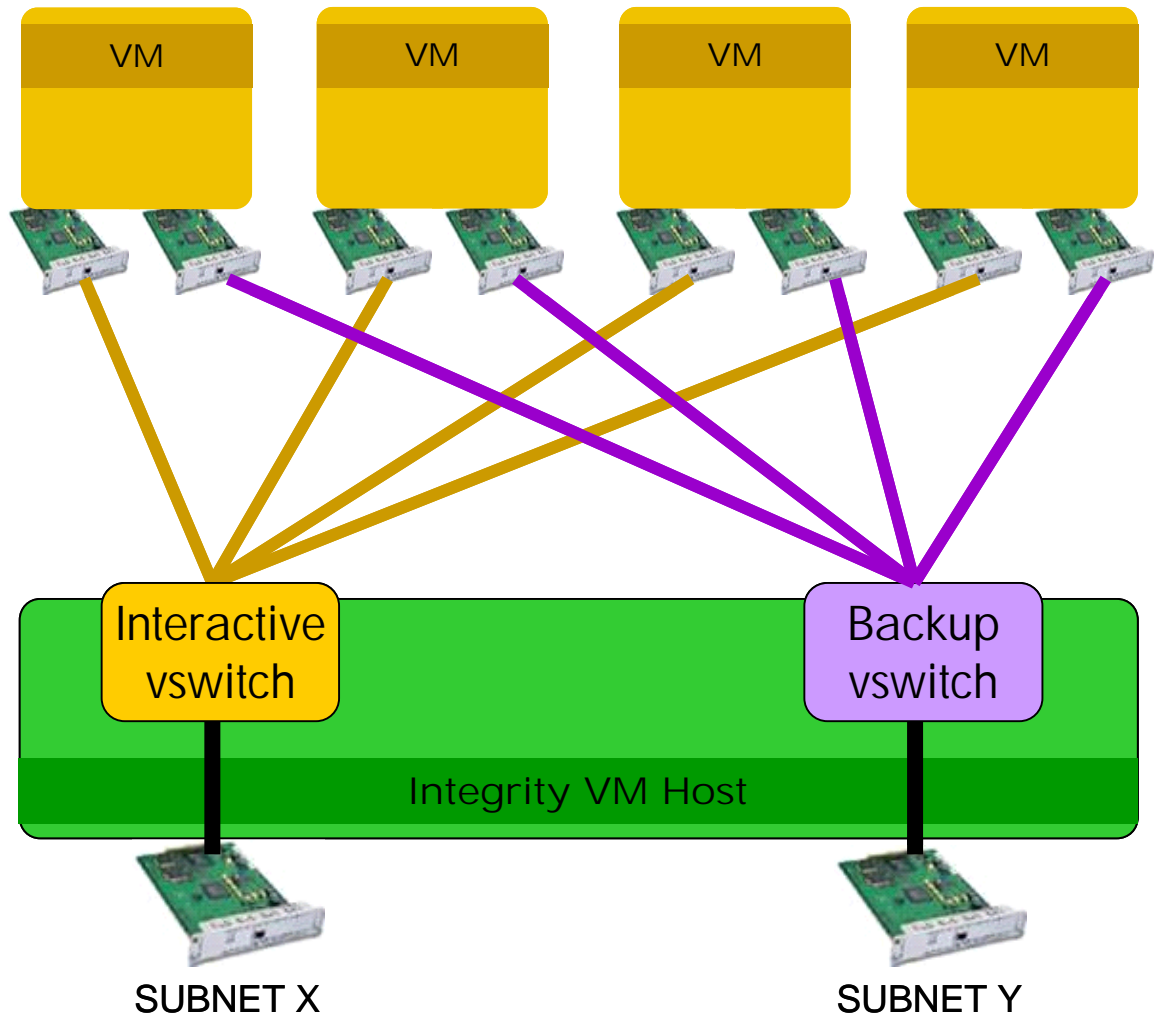


Figure 8 - Leverage existing topologies wherever possible, configure virtual switches to emulate subnet connections for functionality and map virtual NICs accordingly.



Tuning Virtual Machines for Applications

For any given application, the operating system hosting that application may need some tuning so that the application delivers the best functionality and performance for that operating system and server configuration. The specific tuning information is typically provided by the software vendor or, in some cases the hardware (or operating system) vendor.

When deploying an application on a virtual machine, the operating system running on the virtual machine should be tuned for that application as recommended by the software or hardware vendor. The operating on the VM Host should not be tuned for that application – it is already tuned for best VM performance. Moreover, the application is running on the VM's operating system, not that of the VM Host.

For example, suppose a database application is installed on a virtual machine with HP-UX 11iv2. Then the HP-UX installation on the VM should be tuned for that database application as per recommendations for its use with HP-UX 11iv2 on HP Integrity Servers. The VM Host's operating system should not be tuned for the database application – after all, the application is not running on that operating system.

For more information

HP-UX System Management Utilities – www.hp.com/go/systemmanagement

HP's Virtual Server Environment – www.hp.com/go/vse

GlancePlus and other OpenView software – managementsoftware.hp.com/products/gplus

© 2007 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Itanium is a trademark or registered trademark of Intel Corporation in the U.S. and other countries and is used under license.

Windows, Windows Server, and Windows Server2003 are trademarks or registered trademarks of Microsoft Corporation in the U.S. and other countries.

4AA1-1168ENW Rev 2.2, 3/2007

19/19

