HITACHI
Inspire the Next

# Pentaho and Virtual Machines

HITACHI
Inspire the Next

Change log (if you want to use it):

| Date | Version | Author | Changes |
|---|---|---|---|
| 9/10/2017 | 1.0 | Steven Brown | Create |
| | | | |
| | | | |

# Contents

This page intentionally left blank.

# Overview

This document contains information and guidelines around using Pentaho in virtual machines (VMs). It includes information about the configuration of hosts, guest servers, and guest clients.

Our intended audience is Pentaho administrators who are interested in servicing VMs and the Pentaho platform and design tools.

The intention of this document is to speak about topics generally; however, there are advanced features included in each VM platform that should be considered. We do not endorse any single platform but provide guidance on compute patterns known to be pertinent to VMs.

This document applies to these versions of the Pentaho platform:

| Software | Version(s) |
|---|---|
| **Pentaho** | All |

The Components Reference in Pentaho Documentation has a complete list of supported software and hardware.

# Before You Begin

This document assumes that you have knowledge about VMs, and that you have already installed and configured Pentaho.

## *Terms You Should Know*

Here are some terms you should be familiar with:

- **Differencing disk**: A virtual hard drive (VHD) storing changes made to an operating system or VHD, allowing you to isolate those specific changes.
- **Headless server**: A server without access peripherals, such as a mouse or keyboard. It can be used to serve other computers and users.
- **Hypervisor**: Manages a VM.
- **Virtual machine (VM)**: Computer architecture that can emulate computer systems, offering technology to scale past that which is available with simple hardware.

## Use Cases

Use cases employed in this document include the following:

### Use Case 1: The ability to scale past limitations of physical hardware

*By leveraging virtual machines, the ability to scale past limitation of physical hardware are achieved by defining the total of virtual resources greater than the physical resources. Most modern VM platforms offer some elasticity regarding CPU, memory and disk space. This allows the organization to service more needs with less resources.*

### Use Case 2: Provide an isolation level to allocate resources to match compute patterns

*Virtual machines offer the ability to isolate changes, review and either rollback or commit those changes. For example, a large, O/S upgrade is available, but the organization would like to qualify the changes before deploying them to systems that affect developers, user and 3rd parties. In this case a VM may be defined, the software update applied then regression and use case validation may occur. If the results are unacceptable the changes can either be discarded or rolled back. Otherwise, this approach may give the organization additional assurances that the change is safe.*

### Use Case 3: Increase high availability by providing portability of the machine's image

*The virtual machine platform normally can migrate either the VM or the data it services. Depending on the current workload or server availability, this allows for a more consistent user experience as work may be shifted from a heavily burdened server to a less used server or a failed server can be failed-over to a healthy server. With little to no delay this activity can seem transparent to the user or service thereby providing no interruption to service or server requests.*

# Virtual Machines

Virtualization technology can be segmented into the following categories:

*Table 1: Categories of Virtualization*

| Type | Architecture |
|---|---|
| **Virtual machines** | Dedicated instances of an operating system (OS), storage system, network, CPU, and memory. This document provides best practices for setting up and using a VM for work with data integration (DI) and business analytics (BA). |
| **Containers** | Containers sit on top of an OS and share key components such as the kernel, devices, drivers, and other system components, while providing an abstraction layer to an application. Containers are outside the scope of this document. |
| **Cloud** | This is an elastic approach to virtualization where resources can be added or subtracted as workloads fluctuate. Cloud product offerings are also outside the scope of this document. |

There are many VM platforms available in open source and commercial entities from companies such as Oracle, Microsoft, and VMWare. The software is segmented into either a host or guest operating system (OS). A host OS runs on the bare metal, while a guest OS executes inside the VM.

The host hardware and software must support virtualization. Both Intel and AMD support virtualization at the chipset level, whereas Microsoft, CentOS, Red Hat, Ubuntu, and SUSE support virtualization at the OS level. While some providers offer client OS support for virtualization, for production-level efforts, the host OS is normally a server version.

Good planning is key. Because the behavior and expectations between client and server VMs differ, it may be wise to consider separate hosts for each use case.

Guest software can be either a client or server OS depending on the use case. The Pentaho platform is best served by a guest server OS while the Pentaho design tools can be served by a guest client OS.

*Please note that guest clients can never be configured for concurrent shared access.*

# Host Hardware

The host hardware is the foundation of virtualization and should be equipped with enterprise-level components.

You can find details on these topics in the following sections:



## Chipset

To use VMs, start with bare-metal hardware that supports virtualization technology. The host hardware supports virtualization at the chipset level and is controlled by the basic input/output system (BIOS). These features offer the following functional benefits:

- Efficiency in creating VMs
- Efficiencies in switching between VMs for better application responsiveness
- Accelerated performance of VMs by enabling hardware based VM memory management
- Assistance in the live migrations of VMs when the processing load is too much or too little
- Enabling of direct device access of resources by a VM by passing the software layer
- Process isolation for VMs to increase integrity and security
- Nesting of VMs

Hosts should be dedicated to the task of serving either guest clients or guest servers. For production-level environments, it is best to build-out more than one host so that load balancing and failover capabilities are embedded in the solution for scalability and high availability (HA). Refer to your vendor documentation for specific supported typology.

In general, hosts servicing guest clients can be more dynamic depending on your budget. However, processing is consistent and SLA expectations can be met, where guest servers tend to be more static.

## Storage

Many modern VM platforms allow for Storage Area Networks (SAN) or locally attached disk drives. You can configure either with a redundant array of independent disks (RAID), depending on your use case.

*Carefully consider the shared storage component of your solution before building out VMs.*

Segmentation of storage may be based on OS and data. Often, OS images are warehoused where replacement or migration can be performed quickly when needed, whereas data can be duplicated by backups and snapshots at the storage level.

For production-level virtual environments, a SAN is a clear choice for management, point-in-time snapshots, and block-level operations such as migrations. You can configure SAN or attached disk drives for RAID 0 (striped), 1 (mirrored) or 10 (both).

If your data is duplicated elsewhere, you can strip data for performance or mirror the OS for high availability. If your content is not duplicated elsewhere, a stripped and mirrored configuration may be the best alternative.

## Network

The network infrastructure is another host component, which you can configure in one of these ways:

- Create a network that binds the physical network adapter so that VMs can access a physical network.
- Create a network that can be used only by the VMs and the host. This does not allow for connectivity to a physical network.
- Create a network that can be used only by the VMs. This does not allow for connectivity to either a physical network or the host.

For production-level virtual environments, we recommend that you:

- Configure the host networking specifically for virtualization or servers where the backbone is optimized for short hauls on a 10GB backbone.
- Keep the network segments to a minimum. This keeps the source packets closer to the destination.
- Consider other optimization techniques such as acknowledge (ACK) settings and jumbo packets that are designed for heavy server loads without long-distance interference.

## Graphics Processing Unit and Display

Graphics processing unit (GPU) settings can use a pass-through approach or bind to a dedicated GPU. The disadvantage of binding to hardware is the limited number of GPUs available on the host balanced with the number of VMs and their use case.

For the Pentaho platform, host GPU binding may be unnecessary. With other use cases like computer aided design (CAD) and video editing, you may find that selected GPU binding provides better performance.

# Host Operating System

The host OS must support virtualization and, sometimes, nested virtualization where a VM can host a host OS inside a different VM. That compute pattern is outside the scope of this document.

For production-level environments, we recommend installing and configuring server software that supports more than one host, as well as virtualization, to achieve high availability.

- You should also consider a scenario involving automated patching or upgrading so that these machines can be automatically kept up-to-date and adherent to your company's defined policies.
- Ensure the host's profile in the OS is configured for performance. Check all settings, even battery configuration, as some server software comes out of the box configured for balanced use.

Many modern server software supports headless implementations which provide no graphic user interface (GUI) for management and configuration. Instead, you can use scripting or command-line tools, which add complexity and free resources that can be allocated to the VMs.

# Host Hypervisor

After you have installed the OS, install the hypervisor, or VM manager, to manage and configure host-level settings such as some of the hardware described previously in this document.

You can find details on these topics in the following sections:



## Memory Spanning

Memory may be able to span Non-Uniform Memory Architecture (NUMA) nodes to allow VMs to have additional computing resources.

If you use memory spanning in guest server configurations, you may experience degraded performance. We *recommend* avoiding memory spanning in this case.

*Note that memory spanning may not be supported on all host platforms.*

If a host is dedicated to guest clients, this feature may provide scalability greater than the limits of physical resources.

## VM Migrations

Some software allows for migrations of the VM to a rarely-used host physical computer in times of failover, stress, or maintenance.

While documentation may stipulate that this has no impact on VM availability to users, the Pentaho platform is a server-side solution and should be dedicated except for unavailability issues. In that case, a secondary host is the best alternative.

For hosts servicing clients, this feature may stretch your budget or improve the user experience as lesser-used hosts can be placed into service to resource the workload across hosts. As always, carefully consider balance availability with performance.

## Storage Migration

For high availability, load balancing, and maintenance.

The documentation may state that this requires no downtime, but you should use care with this activity as performance may suffer because of the limitations on resources. Therefore, host servers should not be configured to automatically migrate data.

For hosts servicing clients, this feature may stretch your budget or provide a platform for data redundancy for a better user experience. This also requires careful consideration to balance availability with performance.

## Enhanced Session Capability

Some hypervisors feature additional capabilities such as redirection of local devices and resource from computers running a VM connection.

While these capabilities do not impact limited resources, it is always best to configure only what is necessary. For guest servers with little human interaction, this feature has limited benefits and added complexity. Guest clients may benefit from this feature, but your mileage may vary depending on your use case.

# Guest Configuration

Once you have enabled the host hardware and installed the OS and hypervisor software, consider the guest configuration. The host can service either guest client or guest server operating systems.

The Pentaho platform is a server-side solution while the Pentaho design tools are client-side components. You can find details on these topics in the following sections:



## Define and Configure Hard Disks

Before you can create and configure a VM guest, you must define and configure one or more hard disks. Options to decide on include:

### *Disk Format and Type*

**Legacy disk format** supports up to 2GB, while **modern format** supports up to 64TB. There are also **shared disks** which enable backup of VM groups using shared virtual hard disks.

Disk type options may include **fixed size**, **dynamically expanding** or **differencing** with a parent-child relationship between two virtual hard disks.

For guest server configurations, you can achieve greater performance with a modern format with fixed size disk type. The drawback is when you need to expand the storage allocated. At that time, you may be able to reconfigure the hard disk, which takes time, or add a differencing hard disk, which defeats the performance benefits originally configured.

For guest client configurations, you can achieve greater scalability with a modern format and either dynamically expanding, or a combination of fixed and differencing virtual hard disks. Using the latter approach, you could warehouse the OS on a fixed disk and allow for updates to flow to the differencing disk. However, this is not the best performing solution possible.

### *Location*

Consider the location of the virtual hard disks, the place the VHDs are stored on your file system. Depending on how many IO channels are available, it may be best to separate the host OS, host data, guest OS, and guest data on different storage devices. Each should be configured for their specific use case.

## Define Boot Order

Like bare metal BIOS settings, most hypervisors allow for a boot order including disk, optical drive, and network. Security features have matured to include secure boot, which helps prevent unauthorized code from running at boot time. However, secure boot may not be compatible with some OSes.

We recommend that you keep the number of boot devices low and the primary device at the top of the list to speed the boot process. To achieve similar results, but retain the ability to access devices after you mount an image, you can define Virtual CD first, but leave it empty.

## Configure Memory

You can configure memory as dynamic (expanding), weighted (prioritized), or simple allocation (static).

Configure the guest server's memory as static to achieve predictable outcomes. However, if batch processing is your primary use case, you can scale it by allocating resources during the known processing window.

For guest clients, it may make sense to configure for dynamic allocation of memory and even weighted allocation to scale past what is possible with host hardware without virtualization.

## Allocate CPUs

You can also allocate or balance processors among VMs. To achieve predictable outcomes, avoid balancing when allocating. As with memory allocation, one approach to scalability to guest servers is to allocate processing resources during known windows of batch processing.

For guest clients, you may wish to configure resource control of their processors to scale to more users than would be possible without virtualization.

## Choose Disk Controller

Most VM platforms support both integrated development environment (IDE) and small computer system interface (SCSI) controllers. SCSI controllers are a favorite for servers because of their block-level capabilities. IDE drives have been popular with clients for their low cost. Today, SCSI is still a standard for servers, but IDE drives can now perform many of the same functions, including RAID configurations.

Depending on the VM platform, you may not be able to choose a controller. Therefore, the type of virtual hard disk is as important as the controller. When you have a choice, SCSI controllers are a manageable and capable solution for either guest clients or servers.

## Configure Network Adapter

The guest can use the framework of networking once it has been configured. There may be options to limit the bandwidth use.

For guest servers, it is best to limit traffic just from the other VMs. This reduces the attack surface of potentially harmful activities. Do not use bandwidth management if you want to be sure you will consistently meet your service level agreement (SLA). If your use case is primarily batch processing, you can scale by allowing for scheduled startups and stops, allocating your dedicated resources for batch processing without incurring additional hardware expense.

For scalability, guest clients may benefit from bandwidth management so that one user does not consume most of the bandwidth at the expense of others.

## Decide on Enhanced Services

Some VM platforms allow for guests to receive shutdown instructions, synchronize time, generate a heartbeat, or provide backup. Guest clients and servers can use this feature because benefits are achieved with little performance impact, providing safer, improved management.

## Use Checkpoints

Checkpoints provide a snapshot in time where changes can be discarded or merged into the defined virtual hard disk.

*Use care when merging changes, as it is possible to revert your disk to a point in time which does not include* necessary *changes.*

*Table 2: Categories of Virtualization*

| Environment | Checkpoint Usability |
|---|---|
| **Development and Non-Production Guest Server** | May be beneficial. Test changes on the target before you officially implement them. |
| **Production** | Less value in this environment, as the merge operation can be done while the VM is online, but takes some time. Depending on the confidence of the guest server OS, you may want to produce a checkpoint before a major upgrade. Then, you can effectively roll back changes if they do not produce your desired result. |
| **Guest Clients** | Less value for guest clients, as changes should be tested before being deployed. It is best to further segment storage so that snapshots are saved on another device. |

## Configure Paging File

The paging file is the OS-level storage of memory when demand exceeds capacity:

- **Guest servers**: Configure with sufficient memory so that paging files are not used.
- **Guest clients**: Configure the paging file on another storage device for the best user experience.

## Automate Start/Stop Actions

You may be able to instruct the VM platform to shut down a guest when the host is shut down, and automatically restart when the host is reset.

For guest clients and servers in non-production environments, this feature may be beneficial, but it does take significant additional time to shut down or spin up guests if the host needs to be reset more than once.

For production environments, guests should be shut down to avoid riskier operations that may turn off the guest.

# Known Issues and Solutions

Here are a few known issues that you should be aware of, along with best practices and solutions for each issue.

## Poor Performance

Although virtualization has continued to advance, not all problems inherent in it have been solved. Better performance may still be achieved with bare metal, but with the disadvantage of management, scalability, and extendibility. Poor planning, improper configuration, or components not designed for the job may cause problems in the virtual platforms.

### *Solution*

To gain the benefits of VMs in a production environment, load the host with server-quality components. They must have greater capacity than normally required, because they are shared.

Examples include:

- Server-class hardware with a 10GB backbone network
- Ample solid-state drive (SSD) storage for temporal data
- Enterprise-class storage, such as SAN
- Plenty of fast memory

Expect and plan for a significant effort in configuration, monitoring, and tuning the high-end components for specific guest use cases.

## Poor Scalability

If your VM strategy is successful, expect demand to exceed capacity. You may not have enough time to be agile and meet the new demands or growth of a business and its needs. However, VMs can be the basis of your solution.

### *Solution*

Capacity planning is key, as is the communication of the plan to set expectations appropriately. You can achieve VM scalability with certain approaches:

- Equip hosts with enterprise-level components.
- Configure guest clients to scale dynamically.
- Allocate static resources to guest servers but schedule them if the VM platform provides that support.

Here are some other scalability guidelines:

## *16-Core Machines*

Typically, there will be 200 concurrent report requests per 16-core machine. The average response time is 30 seconds. Lower response times require more attention to data modeling, the application of aggregate tables, and the reduction of the number of users per server. The limitation is the IO and the data source response time. To improve the IO, we recommend splitting the 16 cores between a cluster of two 8-core machines (or four 4-core servers), each with 24GB allocated to Pentaho. This is general guidance. The performance you receive will be based on the content, the typical queries, and the performance of the data source.

## *System Monitoring*

You must monitor a system to fine-tune it. When performance is slow, review the system to determine the limiting factors. The less RAM allocated to Pentaho, the more a Java virtual machine (JVM) must page data to and from the disk, slowing down response times. Pentaho cannot predict when a certain amount of GB will not be enough, but you will be able to predict based on your testing and monitoring. Our sizing guidance is based upon allocating 24GB to Pentaho.

## *Scaling*

We recommend horizontal scaling because of typical IO limitations with the amount of traffic between the browser and Pentaho and the data sources. However, vertical scaling is also a viable option, if you make sure enough IO bandwidth is available to support the traffic. In either case, the approach is the same: a cluster of independent servers sharing a single repository. A load balancer would appropriately distribute the load; however, we do require sticky sessions.

## *High Availability and Disaster Recovery Strategies*

The scaling technique described above is also our HA approach. For disaster recovery (DR), your approach will depend on your requirements. The HA strategy could theoretically serve as the DR strategy if some of the clusters are at a different location. However, we typically recommend a cold recovery approach. The repository and environment is backed up regularly so that it can be restored in a new environment within minutes or hours. Our services team can help when you are ready to have this discussion in-depth.

# Practice Example

This example creates a virtual machine on most any platform available.

1. Decide, download and/or purchase a VM platform.
2. Configure the VM platform for shared resources such as input devices, language, display, network and any extensions.
3. Create a virtual hard drive depending on the use case described above. Most often the type of disk drive can't be changed without a conversion effort.
4. Create the virtual machine depending on the user case above. Most of the following settings below can be changed.
5. Configure the virtual machine. Some settings may only be available when the VM has been defined but not yet running such as the number of CPUs, DVD and/or dynamic vs. static resources.
6. Acquire and mount an .ISO image of the guest O/S to the defined DVD drive.
7. Configure the boot order to include the DVD drive.
8. Connect to the virtual machine and start it.
9. Allow the auto install for the guest O/S to complete.

This should result in a fully functional virtual machine. For Linux guest O/S try to connect via SSH or telnet. Since Windows is inherently a GUI O/S continue to connect through the VM platform console. In either case configure the guest O/S as normal with bare-metal hardware and apply all updates.

# Related Information

Here are some links to information that you may find helpful while using this best practices document:

- Pentaho Component Reference
- Best Practices – Pentaho Data Integration Performance Tuning
- Pentaho Performance and Scalability Overview (PDF)
- Virtualization: Top 10 Virtualization Best Practices
- Five easy VM administration tips and virtualization best practices (Requires login)

# Finalization Checklist

This checklist is designed to be added to any implemented project that uses this collection of best practices, to verify that all items have been considered and reviews have been performed.

Name of the Project:_____

Date of the Review:_____

Name of the Reviewer:_____

| Item | Response | Comments |
|---|---|---|
| Did you provision your host with enterprise-level components? | YES_____ NO_____ | |
| Did you plan the host resources sufficiently? | YES_____ NO_____ | |
| Can you allocate resources to a guest server dynamically? | YES_____ NO_____ | |
| Did you allocate resources to client dynamically? | YES_____ NO_____ | |