

RUNNING vtServer IN A VIRTUAL MACHINE ENVIRONMENT



AVT / Vere Technologies
TECHNICAL NOTE

AVT/Vere Technical Note: Running vtServer in a Virtual Machine Environment
Document Revision 1.2 (September, 2016)

© 2016 AVT

vtAlpha and vtVAX are marketed jointly by AVT and Vere Technologies LLC



Table of Contents

1. Introduction..... 4

 1.1. Scope 4

 1.2. Introduction 4

2. General Virtual Machine Considerations 5

 2.1. The Virtualization Decision 5

 2.1.1. Impact of Running on a Virtual Host..... 5

 2.2. Virtual Host Criteria 5

 2.2.1. Performance 5

 2.2.2. Shared Resources 5

 2.2.3. CPU Requirements 6

 2.2.4. USB Port 6

 2.2.5. Storage 6

 2.3. Other Considerations 7

 2.3.1. Symmetric Multiprocessing (SMP)..... 7

 2.3.2. Licensing 8

 2.3.3. vtServer Software Installation 8

3. Hypervisor-specific Information 8

 3.1. VMware 8

 3.1.1. Storage 8

 3.1.2. Network 8

 3.1.3. License Key Considerations 9

 3.2. Hyper-V 2012 or Hyper-V Core..... 9

 3.2.1. Storage 9

 3.2.2. Network 10

 3.2.3. USB License / Network License 10

 3.3. KVM..... 10

1. Introduction

1.1. Scope

This technical note provides details specific to installing vtServer on a virtualized host system, including x86 hypervisor configuration requirements and vtServer installation and operation details.

Procedures common with installations on physical hosts are provided in the *vtServer vtAlpha/vtVAX Reference Manual*.

References to vtServer in this document refer to both vtAlpha and vtVAX for Bare Metal. References to vtVAX apply to the Bare Metal version only. For information on installing vtVAX for Windows in a hypervisor environment, please consult *Running vtVAX for Windows in a Virtual Machine Environment*.

1.2. Introduction

vtAlpha and **vtVAX** emulate the legacy Alpha and VAX hardware platforms, originally developed by Digital Equipment Corporation, on modern 64-bit x86 (x86-64) or compatible platforms. **vtServer** provides the hardware management and user interface common to vtAlpha and vtVAX.

Together, vtServer and the vtAlpha and/or vtVAX emulators serve as a native, or Bare Metal, hypervisor, controlling one or more virtual Alpha or VAX guest systems executing on the same host system. Multiple Alpha and VAX guest systems may execute simultaneously on one host.

The vtServer host may be a physical system or it may be virtualized using VMware, Hyper-V, or KVM.

vtServer installs on Bare Metal: that is, there is no operating system running on the x86 host system. This is similar to other hypervisors such as VMware ESX, Citrix XenServer, and Microsoft Hyper-V. Advantages of the Bare Metal approach over installing as an application under a general purpose operating system include:

- **Maintainability.** There is no external operating system to be updated and maintained separately from the vtServer environment.
- **Reliability.** Only the necessary components are included in the bundled operating environment, which is integrated and tested with the emulation layer. There are no other users on the system to affect performance, stability, or the virtual Alpha/VAX operating environments.

The remainder of this document addresses the considerations unique to the installation and operation of vtServer on a virtualized host system.

2. General Virtual Machine Considerations

This section of the document addresses hypervisor-independent considerations involved with running vtServer on a virtual host. Hypervisor-specific details are addressed in the following sections.

2.1. The Virtualization Decision

vtServer is supported on virtual host systems running under hypervisors listed in the most recent *Host Platform Compatibility List* (HPCL).

We recommend that vtServer be run on a physical host system when possible for the reasons detailed below. The decision to virtualize is often influenced by factors such as local policy and financial considerations that are beyond the scope of this document, which addresses technical issues that directly impact the vtServer software. The decision whether to run vtServer on a virtual host should be made considering both the factors described here and the local considerations.

2.1.1. Impact of Running on a Virtual Host

Placing an x86 hypervisor between vtServer and the physical hardware negates to an extent some of the advantages of the Bare Metal environment noted above:

- **Hardware Control.** The x86 hypervisor has direct control of the physical hardware. It may impose constraints which are not compatible with the way vtServer uses the hardware. Some issues of this nature and workarounds for them are discussed in the sections below.
- **Performance.** The x86 hypervisor itself consumes some system resources. In addition, there are often other guest systems running at the same time as vtServer: if the resources are not allocated properly, as explained below, this can have an adverse impact on the performance, and potentially the stability of the virtual Alpha and VAX systems.
- **Maintainability.** The x86 hypervisor is a software layer that can be updated independent of vtServer. While it is not expected, updates to the x86 hypervisor could have a negative impact on vtServer.
- **Alpha and VAX System Stability.** The vtServer user interface contains safeguards to prevent the emulated Alpha and VAX systems from being shut down unintentionally when the operator is managing the vtServer host system. When the vtServer host is virtualized, the x86 hypervisor operator has ultimate control of the virtual host system, and can shut it down unexpectedly while the virtual Alpha or VAX systems are running.

2.2. Virtual Host Criteria

When configuring the physical host system the x86 emulator will run on, the following details should be considered.

2.2.1. Performance

The maximum achievable performance running vtServer on a virtual host will be somewhat less than running on a physical host using the same hardware platform due to the hypervisor overhead. Typically, I/O performance will be impacted more than CPU performance.

The hypervisor overhead varies considerably depending on a number of factors, such as the hypervisor used, the virtual host configuration, and the nature of the workload. Because of this variability, we are not able to estimate the hypervisor overhead.

2.2.2. Shared Resources

When other guests systems will run on the same x86 host system, CPU, memory, and network resource reservations should be made for the VM used for the vtServer host to ensure that critical hardware resources are always available when needed.

Sharing of these critical resources with other VMs can result in unpredictable performance and possibly unstable operation of the virtual Alpha or VAX system.

2.2.3. CPU Requirements

The rule of thumb is to provide at least 1.5 virtual CPU cores per virtual Alpha CPU or two cores for each virtual VAX CPU; additional cores may improve performance and avoid resource shortages on systems with heavy I/O loads.

Hyper-threaded CPUs should have hyper-threading disabled in the host system BIOS or UEFI to prevent logical threads, which have lower and less predictable performance than the physical cores, from being assigned to an emulation instance. Performance degradation of over 75% has been observed as a result of running vtAlpha/vtVAX on hosts with hyper-threading enabled.

2.2.4. USB Port

If network license servers are not available to provide access to the vtAlpha and vtVAX license information, a dedicated USB port must be provided on the physical host and dedicated to the guest VM that will run vtServer. Please see the Licensing section below and the applicable hypervisor-specific section for considerations and restrictions pertaining to the USB port.

2.2.5. Storage

vtServer supports a variety of storage backends, including storage technologies that were not available on physical VAX or Alpha hardware.

The table below shows the supported storage options for various configurations when running on a physical vtServer host vs. running on a virtual host for both standalone and clustered environments:

	Standalone		Clustered same host		Clustered multi host	
	KZPBA/VTDSK	KGPSA	KZPBA/VTDSK	KGPSA	KZPBA/VTDSK	KGPSA
Local Disk						
Container File	Y	Y	Y	Y	X	X
Physical Disk	Y	Y	Y	Y	X	X
SAN Volume						
Container File	Y	Y	Y	Y	X	X
Physical Disk	Y	Y	Y	Y	P(2); V(2,3)	P(2); V(2,3)
Direct Access	X	P; V(1)	X	P;V(1)	X	P; V(1)
iSCSI Volume						
Container File	Y	Y	Y	Y	X	X
Physical Disk	Y	Y	Y	Y	P(2); V(2,3)	P(2); V(2,3)
NFS Mount Point						
Container File	Y	Y	Y	Y	Y(2)	Y(2)

Legend:

Y – supported on both physical and virtual vtServer hosts
 P – supported on physical vtServer hosts only
 V – supported on virtual vtServer hosts only
 X – unsupported

1 – only FC card with PCI pass-through (VMware)
 2 – cache must be disabled
 3 – only with RAW mapping mode

KZPBA – SCSI adapter
 KGPSA – Fibre Channel adapter

2.3. Other Considerations

2.3.1. Symmetric Multiprocessing (SMP)

When running on multiprocessor configurations, OpenVMS performs frequent checks to ensure that all processors are functioning and that there are no synchronization deadlocks. If abnormal conditions are detected, the system will crash with a CPUSANITY or CPUSPINWAIT bugcheck.

When an emulated multiprocessor Alpha or VAX system is running on a virtual host, certain events, most notably a host migration (e.g., using VMware vMotion), can cause the emulator execution to stall long enough for the SMP sanity timers to expire when set to their default values.

We recommend the following SYSGEN parameter changes to reduce the risk of OpenVMS crashing in such situations:

- SMP_SPINWAIT – increase from 100000 (1 second) to a value between 3000000 (30 seconds) and 6000000 (60 seconds)
- SMP_SANITY_CNT – increase from 300 (3 seconds) to a value between 3000 (30 seconds) and 6000 (60 seconds)

These changes should be added to SYS\$SYSTEM:MODPARAMS.DAT, and then AUTOGEN should be run so they will take effect the next time the system is restarted.

2.3.2. Licensing

The vtAlpha and vtVAX license information is stored on a physical USB license key. In some situations (e.g., the hardware has no available USB port or the x86 hypervisor does not support attaching USB ports to virtual machines) it may not be possible to provide a USB port for the license key on the vtServer host. In high availability and disaster resilient configurations using live host migration features such as VMware vMotion, it is desirable for the license key to be accessible to multiple host systems.

The **vtLicense** server is a standalone network appliance that can host multiple license keys (both production and disaster recovery licenses), making the license content of these keys available across the network. Virtual Alpha and VAX systems can fail over or be migrated between multiple physical hosts without having to move the USB license keys.

More information about the vtLicense server is available on our web site.

2.3.3. vtServer Software Installation

A dedicated (virtual) disk of at least 20GB should be provided for the vtServer software installation. This disk should not be used for storing logical disk container files or other user data to avoid data loss if the vtServer software must be re-installed at a later time.

3. Hypervisor-specific Information

3.1. VMware

Supported versions: ESXi 4.1 and higher.

We discourage the use of a local USB license with VMware ESXi versions prior to 5.5 because VMware occasionally loses connection to USB devices, resulting in license problems.

A vtLicense server is recommended for all VMware environments to facilitate migration of the vtServer host across VMware hosts.

3.1.1. Storage

Supported Storage controllers in VMware are:

- LSI Logic Parallel (Recommended)
- LSI Logic SAS
- VMware Paravirtual

Unsupported Storage controllers are:

- BusLogic Parallel

3.1.2. Network

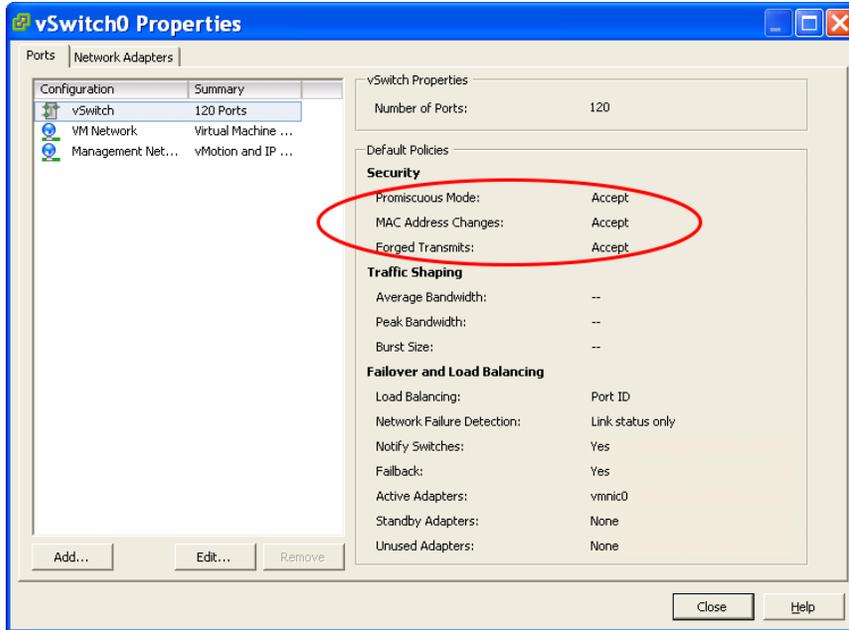
Supported Network adapters in VMware are:

- E1000 (Recommended)
- E1000E
- VMXNET 3

Unsupported Network Adapters are:

- VMXNET 2 (Enhanced)

The adapters used for vtVAX or vtAlpha should have “Promiscuous Mode”, “MAC Address Changes” and “Forged Transmits” enabled on the vSwitch in VMware, as shown below.



3.1.3. License Key Considerations

When running VMware ESX 5.5 or newer, a USB port can be assigned to the vtServer host VM to use for the vtServer license key. This is not recommended with VMware ESX versions prior to 5.5 because the USB device can become detached from the VM, resulting in a loss of the vtServer license with eventual shut down of the vtAlpha or vtVAX emulations if the problem is not corrected within the allowed time (16 hours).

The VMware vMotion feature can be used to improve availability and disaster tolerance of the virtual Alpha and VAX systems by allowing the vtServer host with running emulations to be migrated between VMware host systems. When vMotion is used, we recommend the use of a vtLicense server to provide access to the license keys via the network, removing dependence on a specific host system.

3.2. Hyper-V 2012 or Hyper-V Core

3.2.1. Storage

Supported Storage controllers in Hyper-V are:

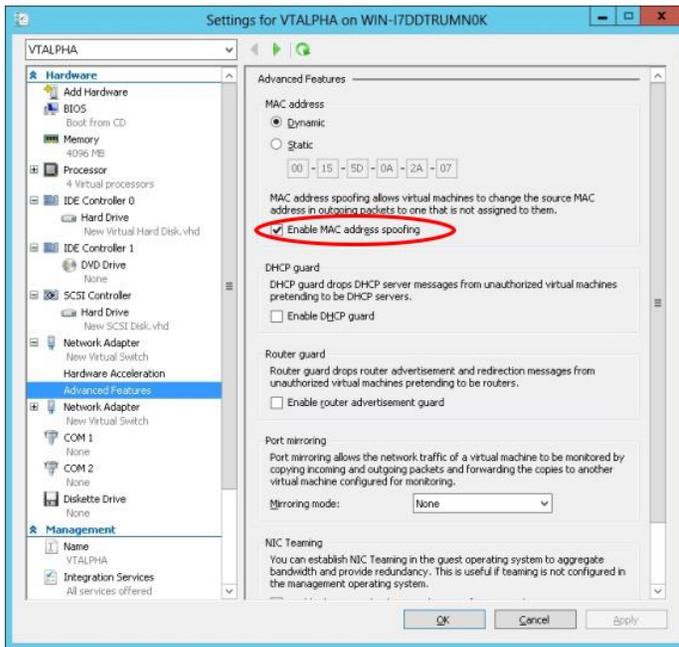
- IDE (Recommended)
- SCSI (only for data, Hyper-V cannot boot from SCSI)

3.2.2. Network

Supported network adapters in Hyper-V are:

- Hyper-V NIC
- Legacy NIC

The adapters used for vtVAX or vtAlpha should have “Mac address spoofing” enabled on the network adapter in Hyper-V.



3.2.3. USB License / Network License

You can use a USB license key directly in Hyper-V 2012 R2 or newer by assigning the USB device to the vtServer VM.

Our recommended way to license vtServer is to use a vtLicense server which has the USB license key connected and serves it over the network. This will also allow Live Migration of vtServer without losing connection to the USB license key.

3.3. KVM

vtServer supports virtio for disks, which may, in some cases, improve disk I/O performance when running vtServer on a virtual host under KVM.