RUNNING vtVAX FOR WINDOWS IN A VIRTUAL MACHINE ENVIRONMENT

TECHNICAL NOTE

WN-0001-05

AVT/Vere Technical Note: Running vtVAX for Windows in a Virtual Machine Environment
WN-0001-05 (October, 2018)

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

Table of Contents

1. Introduction 4
1.1. Scope 4
1.2. Overview 4

2. General Virtual Machine Considerations 5
2.1. The Virtualization Decision 5
2.2. Virtual Host Criteria 5
2.2.1. Performance 5
2.2.2. Shared Resources 5
2.2.3. CPU Requirements 5
2.3. Other Considerations 6
2.3.1. Symmetric Multiprocessing (SMP) 6
2.3.2. Licensing 6

3. Hypervisor-specific Information 7
3.1. VMware 7
3.1.1. Storage 7
3.1.2. Network 7
3.2. Hyper-V 8
3.2.1. Storage 8
3.2.2. Network 8

1. Introduction

1.1. Scope

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

This document supplements the installation procedures provided in the *vtVAX for Windows Installation Guide*.

In this document the term vtVAX applies only to vtVAX for Windows. For information on installing vtVAX for Bare Metal or vtAlpha on a virtual host, please consult *vtServer in a Virtual Machine Environment*.

1.2. Overview

vtVAX emulates the legacy VAX hardware platforms, originally developed by Digital Equipment Corporation, on modern 32-bit or 64-bit x86 or compatible platforms.

vtVAX consists of a management interface and a hypervisor which provides the capability to create one or more virtual VAX guest systems simultaneously on the same host. vtVAX is installed as an application under the Microsoft Windows operating system.

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

It is important for operational stability and security that the vtVAX host system be dedicated to that purpose. Other applications and users should not execute on this system, except as necessary to facilitate vtVAX operations (e.g., server applications such as FTP, SMB, or SSH). Unnecessary services and applications should be disabled or removed.

Of particular note, automatic software updates should be disabled and antivirus, anti-malware and disk defragmentation applications should not be run while the vtVAX emulator is running, as they can consume large amounts of resources and/or block the execution of other applications, including vtVAX, which can lead to unpredictable performance or crashes of the emulator.

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

2. General Virtual Machine Considerations

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

2.1. The Virtualization Decision

vtVAX is supported on virtual host systems running under VMware, Hyper-V, KVM, and Xen hypervisors. For a current list of supported hypervisors and any restrictions, please see the most recent *vtVAX for Windows Product Overview.*

We recommend that vtVAX 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 vtVAX software. The decision whether to run vtVAX on a virtual host should be made considering both the factors described here and any local considerations.

2.2. Virtual Host Criteria

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

2.2.1. Performance

The maximum achievable performance running vtVAX on a virtual host will be somewhat less than that 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 guest systems will run on the same x86 host system, CPU, memory, and network resource reservations should be made for the VM used for the vtVAX 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 VAX system.

2.2.3. CPU Requirements

The rule of thumb is to provide at least 2 virtual CPU cores per 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 vtVAX on hosts with hyper-threading enabled.

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 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

Any host that may be a migration target (e.g., using vMotion) for a virtual machine running vtVAX for Windows must have hardware and operating system configurations that are identical to that of the original host system. * Please note: under Hypervisor guest settings, a static MAC address should be used on every virtual network assigned to a Windows guest. If the MAC address changes from a hypervisor move, then the vtVAX for Windows license will become invalidated.

The vtVAX for Windows license verification is based on a signature that is computed from a number of configuration items: processor manufacturer and type, BIOS revision, number of processors, hard drive signature, OS version, service pack level, and network MAC addresses. If all of these are not identical on the originally licensed host and the migration target, vtVAX will not start until a new license unlock code is obtained. If a migration to a dissimilar host platform must be made, please notify your vtVAX VAR or service provider in advance, if possible, so that provisions may be made to get a timely response to the license update request.

3. Hypervisor-specific Information

vtVAX for Windows has been tested on virtual host systems that were guests under the VMware and Hyper-V versions listed below.

We expect that other hypervisors that support the desired version of Microsoft Windows running in a guest VM will work as well, but this has not been verified by Vere Technologies.

3.1. VMware

Supported versions: ESXi 4.1 and later.

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

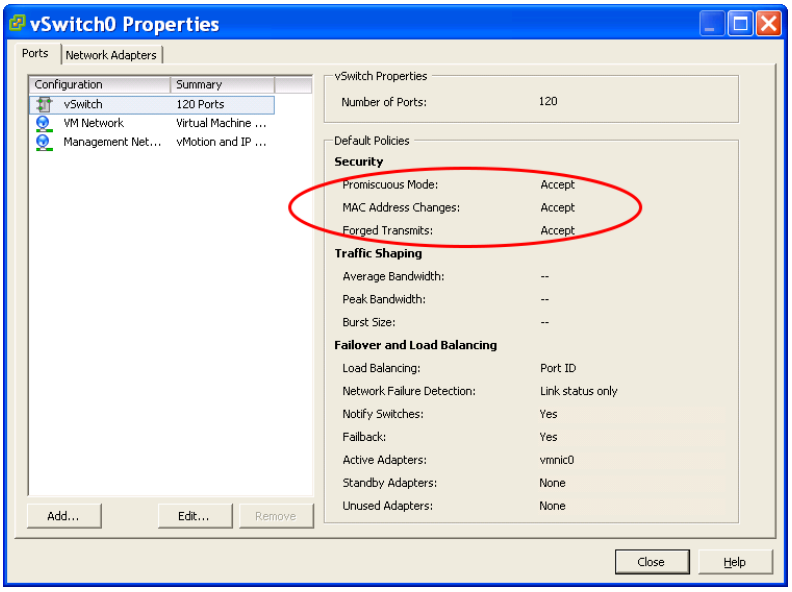
Supported Network adapters in VMware are:

- E1000 (recommended)
- E1000E
- VMXNET 3

Unsupported Network Adapters are:

- VMXNET 2 (enhanced)

The adapters used for vtVAX should have “Promiscuous Mode” and “MAC Address Changes” enabled on the vSwitch in VMware.



3.2. Hyper-V

Supported versions: Hyper-V 2012; Hyper-V Core

3.2.1. Storage

Supported Storage controllers in Hyper-V are:

- IDE (recommended)
- SCSI (data only; 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 should have “MAC address spoofing” enabled on the network adapter in Hyper-V.