

Virtual Iron's Server Virtualization Is Ironclad

27 March 2007

Virtual Iron Software's Virtual Iron builds on the Xen hypervisor and other open-source components to form an effective virtualization solution with a price tag low enough to keep market leader VMware on its toes.

During tests of Virtual Iron 3.5, eWEEK Labs was particularly impressed with the product's provisioning capabilities: We simply plugged our virtualization host servers into a management network, and PXE (Preboot Execution Environment) booted them from Virtual Iron's management server. Once the servers were up, we could begin creating and assigning virtual machines to our nodes right away.

One of the biggest differences between Virtual Iron 3.5 and early Xen-based virtualization products is Virtual Iron 3.5's ability to virtualize pretty much any x86- or x86-64-based operating system - without a special, Xen-aware kernel.

Virtual Iron and other current Xen-based virtualization products manage this modification-free virtualization by running on top of Advanced Micro Devices and Intel processors that feature AMD-V and Intel VT hardware extensions, respectively.

Most notably, this hardware support brings Windows within Xen's ken, a capability that VMware's products have long enjoyed. However, unlike XenSource's XenEnterprise, Virtual Iron 3.5 offers no option for running on hardware without virtualization extensions, which could be a problem if you're hoping to tap virtualization to squeeze more out of your existing machines.

However, the hardware extensions on which Virtual Iron relies are becoming de rigueur for most machines. What's more, after taking into account Virtual Iron's cost advantages over VMware's products, enterprises that choose to go with Virtual Iron might find that they can afford to make some

hardware purchases with their savings.

The full-featured enterprise edition of Virtual Iron 3.5 - which includes live migration, failover and capacity management functionality, as well as support for Fibre Channel SAN (storage area network) and iSCSI storage - costs \$499 per socket. VMware's VI3 Starter is priced similarly - at \$1,000 per pair of CPU sockets - but it lacks support for SAN or iSCSI storage. VMware's VI3 Standard sells for \$3,750 per pair of CPU sockets; VI3 Enterprise, which adds support for VMotion live migration and other high-availability features, costs \$5,750 per pair of CPU sockets.

We tested Virtual Iron 3.5 Enterprise Edition using the free, 30-day trial license. Also available is a free, single-server version of Virtual Iron, in which the management server and virtualization host live on the same machine.

Software

Virtual Iron's management server runs on Linux and Microsoft Windows - specifically, on Red Hat's RHEL (Red Hat Enterprise Linux) 4 U2 (Update 2) and U4 (Update 4) 32- and 64-bit; Novell's SLES (SUSE Linux Enterprise Server) 9 SP (Service Pack) 3 32- and 64-bit; Windows XP Professional 32-bit; and Windows Server 2003 32-bit.

eWEEK Labs tested the Virtual Iron management server, which is Java-based, on Windows Server 2003 SP2 and on CentOS 4.2 (a clone of RHEL 4 U2).

Also cross-platform-friendly is Virtual Iron's Management Console, a fairly rich application that depends on Sun Microsystems' Java Web Start application technology.

Using the console, we were able manage our nodes equally well from Windows and from Linux - something we found particularly refreshing after

coming off our testing of VMware's Virtual Infrastructure product, which is disappointingly Windows-bound on both its management server and client application sides.

Virtual Iron can host pretty much any x86 or x86-64 operating system, but there's a significant set of functionality that requires add-on software, called VS Tools, that Virtual Iron makes available only for a handful of operating systems (the same Windows and Linux versions that are on Virtual Iron's list of supported management servers).

VS Tools are required for Virtual Iron's LiveMigrate and LiveRecovery features, to view performance information on guest instances from the management console, and to shut down or reboot guest machines gracefully (as in not simply pulling their virtual plugs) from the management console.

For Windows guests, these tools take the form of an .exe installer with Virtual Iron-enhanced device drivers. For Linux, Virtual Iron provides binary packages containing drivers compiled to match supported kernels.

We'd like to see Virtual Iron adopt a less rigid approach to delivering these tools for Linux. VMware, for example, allows administrators to compile the drivers to match the kernel they're running, rather than limit support to a handful of options.

This sort of flexibility would have helped prevent one of the snags we encountered during testing: We installed the x86-64 version of CentOS 4.4, and, probably because the VM we'd created sported a single processor, the CentOS 4.4 installer installed the uniprocessor version of the Linux kernel. Virtual Iron's VS Tools packages come in SMP (symmetric multiprocessing) flavors only, and an SMP kernel must be installed to install the tools properly.

When we installed the SMP kernel for our system from its network repository, however, the version we pulled down was different from the version that the VS Tools package was expecting, so we had to revert to the older kernel for the tools to work.

We were surprised to find that Virtual Iron does not offer its VS Tools for rPath Linux because Virtual Iron and rPath have a partnership in which rPath has added Virtual Iron as a build option for software appliances created with the rBuilder platform.

During our tests, we downloaded an rPath-based MediaWiki appliance in Virtual Iron format, dropped the appliance's virtual disk into the appropriate folder in our Virtual Iron management server and assigned the disk to a new VM. Without VS Tools support, however, the virtual appliance was significantly less useful.

If pressed, we probably could have adapted the supported Red Hat kernel to run our MediaWiki appliance, but we'd rather see Virtual Iron take care of that.

Management

It wasn't too tough to create new VMs using Virtual Iron's Management Console, but the process is definitely rougher around the edges than that of VMware's virtualization products. For one thing, it's necessary to visit different parts of the console to configure a VM's CPU and RAM settings, its network adapters, and its virtual disks.

During our tests of VI3, we connected our VMware ESX servers to the FTP server on which we store, among other things, operating system installation images. We could then attach these images to VMs we'd created as virtual CD or DVD drives, install from those images, and then access their contents once our machines were installed.

With Virtual Iron, the VM creation interface sports a handy drop-down menu of available installation images, but these images had to reside in a particular folder on our management server to show up on the list. This would have meant copying images from our standard FTP store to that particular server.

We ended up dumping the Windows Server 2003 machine that we'd initially chosen to host the management server in favor of a CentOS 4.2 server with our OS image store mounted as a Sun NFS (Network File System) share. We then

symlinked the iso images we wanted to use to the requisite Virtual Iron directory.

This was not a tough workaround, since we'd planned on trying out the management server on both Windows and Linux hosts anyway, but we'd like to see future Virtual Iron versions develop more flexible access to storage.

We could access and control our VMs through console windows that we launched from the management interface. With virtual instances for which we'd installed VS Tools, we could power cycle, reboot or shut down the VMs, but we could not pause them, which is something we're accustomed to being able to do with other virtualization products.

We also missed having a snapshotting functionality similar to what VMware offers, but we could clone our virtual disks and later replace a machine's disk with a clone, thereby restoring to an earlier point in time.

Virtual Iron's LiveMigrate feature worked fine for our guests with VS Tools installed and with disks stored on our iSCSI appliance: We just dragged the VMs from one node to the other and hit the confirm button. Each migration took less than 15 seconds to complete.

Our experience with Virtual Iron's LiveRecovery feature wasn't so smooth. We tried yanking the power cord from one of our nodes that was hosting the Windows Server 2003 and CentOS guests, and the management server told us that it wasn't attempting an autorecovery because the node "may be still active."

We then tried disconnecting one of our nodes from the management server, but this didn't trigger an autorecovery, either. It turns out that we were bumping up against safeguards that prevent so-called "split brain" scenarios, and we didn't have a chance to sort out these issues before the end of our testing.

Hardware

As mentioned earlier, Virtual Iron requires server

hardware with AMD-V or Intel VT hardware extensions for its host nodes.

The management server doesn't require any particular processor type, but redundancy and fast I/O is important for the management server because the nodes depend on it. Virtual Iron 3.5 supports a maximum of 32 CPUs and 96GB of RAM per node, and the product can expose as many as eight CPUs to its guest machines.

We tested Virtual Iron 3.5 on a pair of Dell PowerEdge 430 servers with Intel 3GHz Pentium D processors and 2GB of RAM each. Each machine sported three NICs - one for the management network, one for an iSCSI network, and one for accessing the Internet and other servers in our environment. (Virtual Iron maintains a hardware compatibility list for its products here .)

Virtual Iron's iSCSI support is new in Version 3.5, and the list of supported iSCSI hardware is somewhat slim at this point. We thus turned to the same do-it-yourself Openfiler-based iSCSI target with which we recently tested VI3.

After some initial troubles in setting our network configuration, we were able to access the volume we created in Openfiler for use with Virtual Iron, slice it up into disks and install VMs without further trouble. We also could install VMs in the disks local to each of our nodes, but we could not use LiveMigrate with machines configured in this way.

Copyright 2007 by Ziff Davis Media, Distributed by United Press International

APA citation: Virtual Iron's Server Virtualization Is Ironclad (2007, March 27) retrieved 27 November 2020 from <https://phys.org/news/2007-03-virtual-iron-server-virtualization-ironclad.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.