



Java density performance comparison: Red Hat Kernel Virtual Machine with KSM vs. Microsoft Hyper-V on the Dell PowerEdge R710 solution

Executive summary

Red Hat, Inc. (Red Hat) commissioned Principled Technologies (PT) to compare virtualized Java workload density performance of the new Red Hat® Kernel Virtual Machine® (KVM) technology using Kernel Samepage Merging (KSM) to Microsoft® Windows Server® 2008 R2 with Hyper-V™ technology (Hyper-V) running on the same server-and-storage solution. We compare the following two solutions:

- Red Hat KVM running on Intel® Xeon® Processor X5550-based Dell™ PowerEdge™ R710 server with 24 GB of memory (Red Hat KVM solution)
- Microsoft Hyper-V running on Intel Xeon Processor X5550-based Dell PowerEdge R710 server with 24 GB of memory (Microsoft Hyper-V solution)

KEY FINDINGS

- Using KSM to manage memory allocation, the Red Hat KVM solution ran 32 VMs on the Java workload performance. (See Figure 1.)
- The Red Hat KVM solution delivered four times as many VMs as the Hyper-V solution, 32 VMs and 8 VMs, respectively. (See Figure 2.)

The Dell PowerEdge R710 server had two 2.66GHz Intel Xeon Processor X5550s and 24 GB of RAM. For the Red Hat KVM configuration, we installed Red Hat Enterprise Linux 5.4 Beta (kernel-2.6.18-159.el5) as the host operating system, and installed Windows Server 2008 SP2 32-bit as the guest operating system. For the Microsoft Hyper-V configuration, we installed Microsoft Windows Server 2008 R2 running with Hyper-V enabled, and installed Windows Server 2008 SP1 32-bit as the guest operating system. The Red Hat KVM solution offers KSM, which is Linux kernel code that scans system memory, looking for memory duplication across similar or identical applications. KSM enables the Red Hat KVM solution to improve Java workload density performance by reallocating and managing system memory, so that the Red Hat KVM solution can dynamically share this memory, use it more efficiently, and oversubscribe it to a greater number of VMs than it would otherwise support. Each virtual machine (VM) uses 3 GB of RAM. The Microsoft Hyper-V solution does not offer anything similar to KSM.

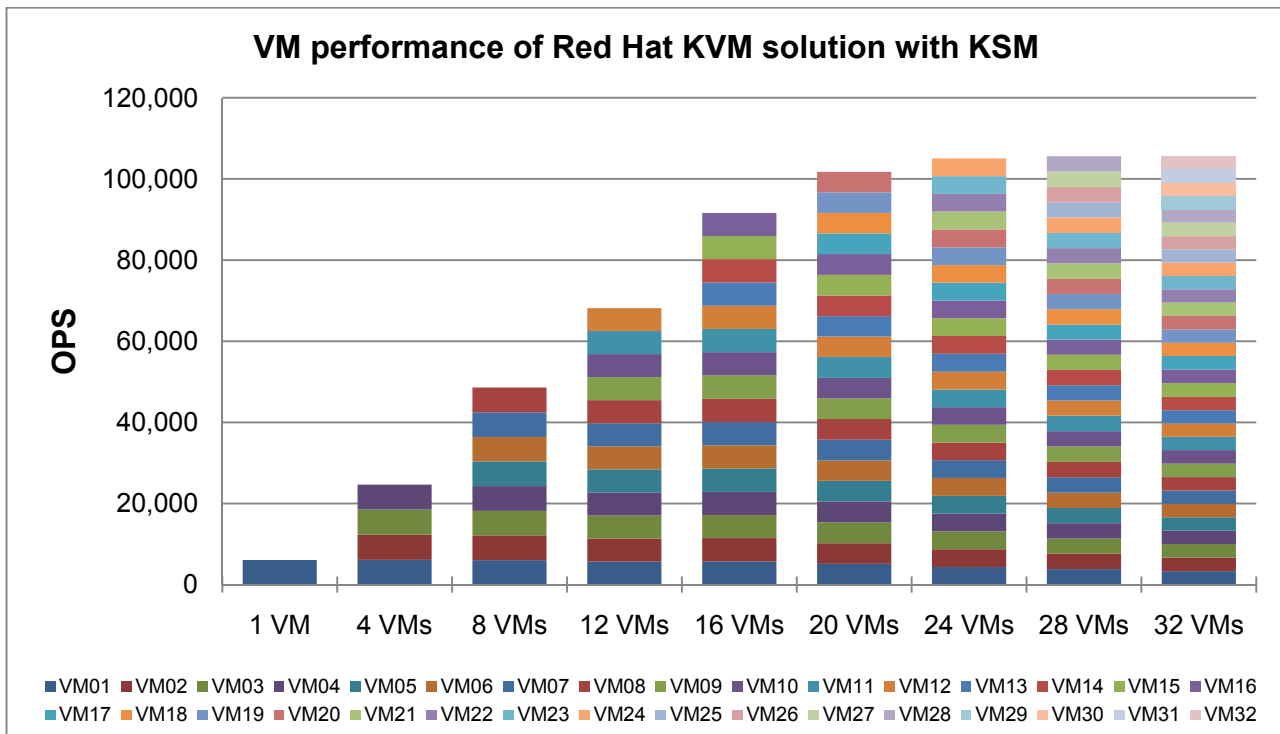


Figure 1: Java density performance of the Red Hat KVM solution with KSM running multiple VMs. Higher numbers are better.

For this comparison, we selected a Java benchmark, whose reporting metric is operations per second. We present more information about this benchmark in our Workload section.

Figure 1 shows the Java workload density performance of the Red Hat KVM solution with KSM running. By allowing the Red Hat KVM solution to oversubscribe its memory four times over, KSM enables the Red Hat KSM solution to run 32 VMs (which Red Hat refers to as “guests”) from 24 GB of RAM. At 32 VMs, the CPU utilization approaches 100 percent.

Figure 2 shows the median operations per second (OPS) for the Red Hat KVM solution with KSM and the Microsoft Hyper-V solution. On the Dell PowerEdge R710 server with 24 GB of RAM, the Red Hat KVM solution with KSM ran 32 VMs, a maximum of 105,662 OPS, while the Microsoft Hyper-V solution, which does not have KSM, ran 8 VMs, a maximum of 46,196 OPS. Each result is the median score of three runs of the benchmark. (See the Test results section for the scores from all three runs.) A higher OPS score indicates the solution is able to handle more Java requests and thus deliver greater throughput.

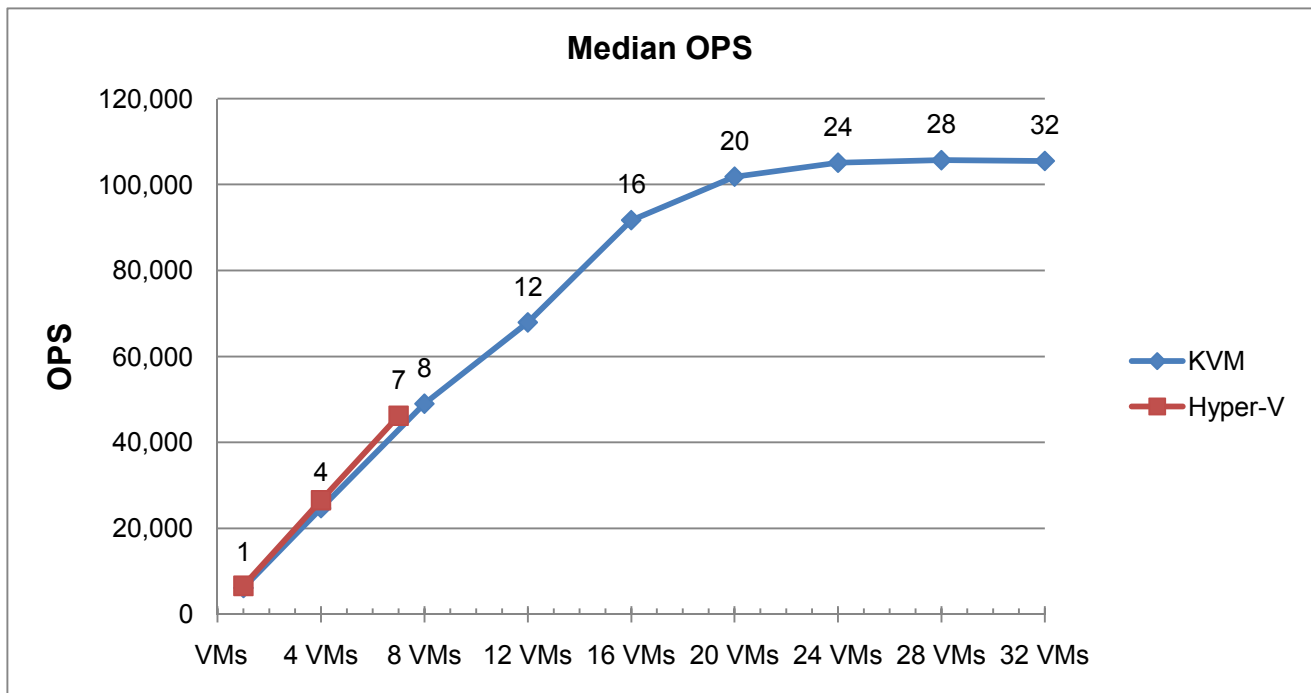


Figure 2: Median OPS results for the Red Hat KVM solution with KSM and the Microsoft Hyper-V solution. Higher numbers are better.

Workload

We used a custom Java benchmark to measure the server’s Java performance. The benchmark models the three-tier client/server architecture, with the middle layer as the primary focus.

The Java application workload utilizes multiple data groups and multiple threads as it runs. Each data group consists of roughly 25 MB of data objects. Each thread represents an active user posting transaction requests within the data group. We configured the benchmark to use six data groups and throttled the throughput so that a VM vCPU was not completely saturated. The goal was to have the processor utilization inside the VM average between 50 and 60 percent with only one VM running.

Test results

Before we started the Java benchmark, we logged into the system, started each server, and allowed the servers to sit for 15 minutes before we started the run.

Figure 3 shows the OPS score for each solution running different numbers of VMs the Red Hat KVM solution with KSM ran a maximum of 32 VMs, while the Microsoft Hyper-V solution ran a maximum of 7 VMs.

	Red Hat KVM solution with KSM	Microsoft Hyper-V solution
1 VM OPS	11,033,251	11,945,822
4 VMs OPS	44,449,704	47,774,201
7VMs* OPS	NA	83,152,695
8 VMs OPS	88,187,982	Unable to run
12 VMs OPS	122,250,238	Unable to run
16 VMs OPS	165,091,261	Unable to run
20 VMs OPS	183,294,679	Unable to run
24 VMs OPS	189,147,463	Unable to run
28 VMs OPS	190,191,751	Unable to run
32 VMs OPS	189,871,385	Unable to run

Figure 3: Java benchmark results, in JVM, for the Red Hat KVM solution with KSM and the Microsoft Hyper-V solution. Higher numbers are better

* Microsoft Hyper-V solution only

Test methodology

Adjusting BIOS settings

We enabled the virtualization feature in the BIOS for our virtualization testing.

Setting up the host server with Red Hat 5.4 Beta with RHEL KVM

For the Red Hat KVM configuration, we installed kernel-2.6.18-159.el5 of Red Hat Enterprise Linux 5.4 Beta on the host server.

Installing Red Hat 5.4 Beta

1. Insert and boot from the Red Hat Enterprise Linux 5.4 install DVD.
2. Press Enter to install using graphical mode.
3. At the media test screen, select Skip.
4. At the Red Hat Enterprise Linux 5 title screen, click Next.
5. At the Choose a Language screen, select English, and click Next.
6. At the Keyboard Type screen, select U.S. English, and click Next.
7. At the Installation Number pop-up screen, enter the Installation Number for Virtualization, and click OK.
8. At the Installation requires partitioning screen, click Next.
9. If a warning appears, click Yes.
10. At the hard drive layout screen, click Next.
11. At the Boot loader screen, click Next.
12. At the Network Devices screen, click Edit, and deselect Enable IPv6. Click OK.
13. Under Hostname, select Manually, and set the hostname to `##.domain.com`, where `##` is the name of your host server. Click Next. If you receive any warnings, click Continue.
14. At the Time zone selection screen, select the appropriate time zone, and click Next.
15. Enter the root password in the root password and confirm fields, and click Next.
16. At the default installation screen, leave the customized items unchecked, and click Next.
17. Click Next to begin the installation.
18. At the Congratulations screen, click Reboot.
19. After the system reboots and the Welcome screen appears, click Forward.
20. Agree to the EULA, and click Forward.
21. At the Firewall screen, select Disabled from the Firewall drop-down menu, and click Forward.

22. If a warning pop-up screen appears, click Yes.
23. At the SELinux screen, select Disabled from the SELinux Setting drop-down menu, and click Forward.
24. If a warning appears, click Yes.
25. At the Kdump screen, uncheck Enable Kdump, and click Forward.
26. At the Date and Time screen, set the time and date, and click Forward.
27. At the Set Up Software Updates screen, select No, I prefer to register at a later time, and click Forward.
28. If a pop-up screen appears asking if you are sure, click No thanks.
29. At the Finish Updates Setup, click Forward.
30. At the Create User Screen, click Forward.
31. If a warning appears, click Continue.
32. At the Sound Card screen, click Forward.
33. At the Additional CDs screen, click Finish.
34. At the pop-up screen, click OK.
35. At the logon screen, type `root` for the user and `Password1` for the password, and press Enter.

Installing the KVM packages in RHEL 5.4 Beta

1. Copy the RHEL 5.4 ISO image, and mount it at `/mnt/rhel54/`
2. Create a text file at `/etc/yum.repos.d/rhel-54-dvd.repo` with the following contents:

```
[rhel-54-beta-dvd]
name=Red Hat Enterprise Linux $releasever - $basearch - DVD
baseurl=file:///mnt/rhel54/Server/
enabled=1
[rhel-54-vt-beta-dvd]
name=Red Hat Enterprise Linux $releasever VT - $basearch - DVD
baseurl=file:///mnt/rhel54/VT/
enabled=1
```

3. Type the following command to install KVM packages:

```
# yum groupinstall kvm
```

Setting up the network adapters

1. Log into the host, right-click the desktop, and choose Open Terminal.
2. Type the following command to edit the network configuration settings, where X is the relevant host NIC to modify:

```
# gedit /etc/sysconfig/network-scripts/ifcfg-ethX &
```

3. Modify the following lines to set the static IP address and netmask:

```
BOOTPROTO=static
IPADDR=192.168.XXX.XXX
NETMASK=255.255.255.0
```

4. Save the file, and exit gedit.
5. Type the following command to modify the remaining network settings:

```
# gedit /etc/sysconfig/network &
```

6. Modify the following lines to disable IPv6 and set the hostname:

```
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=hostname.local.domain
```

7. Save the file, and exit gedit.
8. Reboot the server.

Configuring additional tuning options in the host

Create a shell script to disable the following services:

```
service auditd stop
service avahi-daemon stop
service anacron stop
service irqbalance stop
service qpidd stop
service smartd stop
service crond stop
service haldaemon stop
service opensmd stop
service openibd stop
service yum-updatesd stop
service collectd stop
service bluetooth stop
service cups stop
service hidd stop
service isdn stop
service kudzu stop
service mcstrans stop
service mdmonitor stop
service messagebus stop
service restorecond stop
service rhnsd stop
service rpcgssd stop
service setroubleshoot stop
```

Installing the systat package for system monitoring

1. Insert the RHEL 5.4 DVD into the host machine.
2. Browse to the DVD drive.
3. Find the systat package under the Server directory, labeled systat-7.0.2-3.el5.x86_64.rpm
4. Install the package by typing the following command:

```
# rpm -i systat-7.0.2-3.el5.x86_64.rpm
```

Setting up the host server with Microsoft Windows Server 2008 R2 Hyper-V

For the Hyper-V, we installed Windows Server 2008 R2 and enabled the Hyper-V role.

Installing Windows Server 2008 R2

1. Boot the server, and insert the Windows Server 2008 R2 installation DVD in the DVD-ROM drive.
2. At the Language Selection Screen, click Next.
3. Click Install Now.
4. Select Windows Server 2008 Datacenter (Full Installation) x64, and click Next.
5. Click the I accept the license terms check box, and click Next.
6. Click Custom.
7. Click Drive options (advanced).

8. Delete any existing partitions.
9. Ensure you have selected the first drive, and click New.
10. Click Apply.
11. Click Continue.
12. Click Next.
13. At the User's password must be changed before logging on warning screen, click OK.
14. Type `Password1` as the new password in both fields, and click the arrow to continue.
15. At the Your password has been changed screen, click OK.

Installing the Hyper-V Role in Windows Server 2008

1. Click Start→Administrative Tools→Server Manager.
2. Click Roles.
3. Click Add Roles.
4. Click Next.
5. Select the Hyper-V role by checking the box beside it, and click Next.
6. Click Next.
7. Choose the NIC you wish to use as your virtual network.
8. Click Install to begin the Hyper-V installation.
9. When the installation finishes, click Close.
10. When the installer prompts you to restart now, click Yes.
11. Follow the prompt to log onto the system.
12. Enter the password, and click the arrow to continue. The Resuming Configuration Wizard should start automatically.
13. Once the Resuming Configuration Wizard completes, the installation results should appear with a message that the installation succeeded.
14. Click Close.

Creating and configuring the VMs on Hyper-V and RHEL 5.4

Creating the virtual machine with Hyper-V Manager (Windows Server 2008)

1. Click Start, Administrative Tools, Hyper-V Manager.
2. Click New, Virtual Machine.
3. Click Next.
4. Enter a name for the virtual machine, such as VM1, and click Next.
5. Enter 3072MB for the VM memory, and click Next.
6. Choose the virtual network, and click Next.
7. Enter 15GB for the virtual hard disk size, and click Next.
8. Click Finish.
9. Insert the installation DVD for Windows Server 2008 SP2 32-bit into the DVD drive.
10. In Hyper-V Manager, right-click the virtual machine, and choose connect.
11. On the media menu, choose DVD Drive→image file.
12. Browse to the Windows Server 2008 SP2 32-bit ISO location.
13. Locate the ISO file, select it, and click Open.
14. On the Action menu, choose Start.

Creating the virtual machine using the Virtual Machine Manager (RHEL 5.4)

1. Log into the RHEL 5.4 host.
2. Click Applications, System Tools, Virtual Machine Manager.
3. Click New.
4. Click Forward.
5. Choose a name for the VM. Click Forward.
6. Keep the default of "Fully virtualized," choose the CPU architecture, and choose KVM as the hypervisor.
7. Assuming you have copied the RHEL 5.4 beta media to the host machine, choose "Local install media," Windows as the OS Type, and Windows Server 2008 as the OS Variant. Click Forward.
8. Click Browse to browse to the Windows Server 2008 SP2 32-bit ISO location.
9. Locate the ISO file, select it, and click Open. Click Forward.

10. Choose File (disk image), and specify the location where you wish to store the image file. In our case, we chose /images/VM#.IMG.
11. Specify 15000MB for the size of the image file, and uncheck the “Allocate entire virtual disk now” checkbox. Click Forward.
12. Choose “Shared physical device,” and select the physical NIC configured for virtual networking. Click Forward.
13. Set Max memory size and Startup memory size to 3072MB, and set the number of Virtual CPUs to 1. Click Forward.
14. Review the summary information, and click Finish.

Installing the guest operating system (both platforms)

1. Choose the language, time and currency, and keyboard input. Click Next.
2. Click Install Now.
3. Choose Windows Server Enterprise (Full Installation). Click Next.
4. Accept the license terms, and click Next.
5. Click Custom.
6. Click the Disk, and click Drive options (advanced).
7. Click New, Apply, Format, and then click Next.
8. Let the installation process continue. The VM will reboot several times.
9. After the installation completes, click OK to set the Administrator password.
10. Enter the administrator password twice, and click OK.

Installing Integration Components (Hyper-V)

1. On the Hyper-V manager Media menu, choose Action→Insert Integration Services Setup Disk.
2. Shut down the VM after Integration Services install completes.

Creating the additional VMs on the host server

After installing and configuring the initial VM, we created the additional VMs by using virt-clone on the RHEL platform and manual creation of new VMs on the Hyper-V platform.

Cloning the VMs on RHEL

Run the following command to make a clone of VM1 you name VM2. Repeat as necessary for multiple VMs.

```
# virt-clone --connect qemu:///system --original VM1 --name VM2 --file /images/VM2.IMG
```

Cloning the VMs on Hyper-V

1. In Hyper-V Manager, right-click VM1, and choose Settings.
2. Remove the virtual hard drive from the IDE controller.
3. Create a new VM with the same specifications as VM1.
4. Copy and paste all virtual hard drives to the location reserved for the new VM.
5. Reattach the virtual hard drive to the new VM's IDE controller.
6. Start the new VM.
7. Repeat steps 1 through 6 for remaining VMs.

Configuring the additional VMs after cloning

Modify the IP Address and Computer Name for each VM:

1. Power on the VM, and open the console.
2. Click Start, Control Panel, Network Connections, and click Manage Network Connections.
3. Right-click the NIC, and choose properties.
4. Select TCP/IP (v4), and choose properties.
5. Set the IP address, subnet, gateway, and DNS server for this NIC, which will handle outgoing server traffic (e.g., the public IP). Click OK, and click Close.
6. Click Start, Control Panel, and double-click System.
7. Click Change Settings.

8. Click Change.
9. Enter the new computer name, and click OK.
10. Click OK to restart, click Close, and click Restart Now.
11. Repeat steps 1 through 10 for remaining VMs.

Testing procedure

To perform the test, we used a series of batch files and shell scripts, executed from a controller machine. We stored batch files each system to control the benchmark settings and we used the PsExec utility to coordinate the test procedures amongst the virtual machines. We use simple windows file sharing to copy files from machine to machine, as we needed.

The testing procedure consisted of the following steps:

1. Start each VM one at a time, making sure the previous VM completely loaded before starting the next one.
2. Pause for 15 minutes to wait for background tasks to complete before server reboot.
3. Start performance statistics gathering processes on client machines and server machine.
4. Shutdown the virtual machines.
5. Start the VMs under test.
6. Wait 2 additional minutes for any background tasks to complete.
7. Start performance statistics gathering processes on client machines and server machine.
8. Start the workload connections.
9. Start the workload ramp up period.
10. Start the workload.
11. Stop the workload.
12. Copy all output files to the controller.
13. Shutdown the virtual machines.
14. Reboot the server under test.
15. Wait for a ping response from the physical machine.

Java configuration

The Java workload requires a Java Virtual Machine on the system under test. We used the Sun Java SE 6 Run Environment (Build 1.6.0_16-b01) and left the default installation settings.

We created a shell script, which we placed in the root Java workload directory, to issue the Java run command to launch the benchmark. We changed to the Java workload root directory and executed the shell script from that directory to start testing.

In the script file, we set the Java options that control the performance of the JVM:

- `-Xmx1550m` sets the maximum heap size.

KSM configuration

We used the following script to control KSM on RHEL 5.4:

```
#!/bin/sh

# stand-alone bash version of vdsm's ksm.py
# monitor system memory load and tune ksm accordingly
#
# needs testing and ironing. contact danken@redhat.com if something breaks.
DEBUG=1
KSMCTL="ksmctl"

KSM_MONITOR_INTERVAL=${KSM_MONITOR_INTERVAL:-60}
KSM_NPAGES_BOOST=${KSM_NPAGES_BOOST:-300}
KSM_NPAGES_DECAY=${KSM_NPAGES_DECAY:--50}
```



```

KSM_NPAGES_MIN=${KSM_NPAGES_MIN:-64}
KSM_NPAGES_MAX=${KSM_NPAGES_MAX:-1250}
# microsecond sleep between ksm scans for 16Gb server. Smaller servers sleep
# more, bigger sleep less.
KSM_SLEEP=${KSM_SLEEP:-10000}

KSM_THRES_COEF=${KSM_THRES_COEF:-30}
KSM_THRES_CONST=${KSM_THRES_CONST:-2048}

total=`awk '/^MemTotal:/ {print $2}' /proc/meminfo`
[ -n "$DEBUG" ] && echo total $total

npages=0
sleep=$((KSM_SLEEP * 16 * 1024 * 1024 / total))
[ -n "$DEBUG" ] && echo sleep $sleep
thres=$((total * KSM_THRES_COEF / 100))
if [ $KSM_THRES_CONST -gt $thres ]; then
    thres=$KSM_THRES_CONST
fi
[ -n "$DEBUG" ] && echo thres $thres

committed_memory () {
    # calculate how much memory is committed to running qemu processes
    local progname
    progname=${1:-qemu}
    ps -o vsz `pgrep $progname` | awk '{ sum += $1 }; END { print sum }'
}

increase_napges() {
    local delta
    delta=${1:-0}
    npages=$((npages + delta))
    if [ $npages -lt $KSM_NPAGES_MIN ]; then
        npages=$KSM_NPAGES_MIN
    elif [ $npages -gt $KSM_NPAGES_MAX ]; then
        npages=$KSM_NPAGES_MAX
    fi
    echo $npages
}

adjust () {
    local free committed
    free=`awk '/^MemFree:/ { free += $2}; /^Buffers:/ {free += $2}; /^Cached:/
{free += $2}; END {print free}' /proc/meminfo`
    committed=`committed_memory`
    [ -n "$DEBUG" ] && echo committed $committed free $free
    if [ $[committed + thres] -lt $total ]; then
        $KSMCTL stop
        [ -n "$DEBUG" ] && echo "$[committed + thres] < $total, stop ksm"
        return 1
    fi
    [ -n "$DEBUG" ] && echo "$[committed + thres] > $total, start ksm"
    if [ $free -lt $thres ]; then
        npages=`increase_napges $KSM_NPAGES_BOOST`
        [ -n "$DEBUG" ] && echo "$free < $thres, boost"
    else

```

```

        npages=`increase_napges $KSM_NPAGES_DECAY`
        [ -n "$DEBUG" ] && echo "$free > $thres, decay"
    fi
    $KSMCTL start $npages $sleep
    [ -n "$DEBUG" ] && echo "$KSMCTL start $npages $sleep"
    return 0
}

usage () {
    echo usage:
    echo "  $0 [-d]"
    echo
    echo inform $0 of qemu-kvm exec/death with
    echo '  pkill -P `pgrep ksm-monitor` sleep'
    exit 1
}

daemon () {
    while true
    do
        sleep $KSM_MONITOR_INTERVAL
        adjust
    done
}

case x"$1" in
    x-d) daemon & echo ksm monitor pid: $! ;;
    x) adjust ;;
    *) usage ;;
esac

```

Appendix A – Server configuration information

This appendix provides detailed configuration information about the two Dell PowerEdge R710 servers. Figure 4 provides detailed system configuration information. Note that, with the exception of the OS, the configuration of the Dell PowerEdge R710 running the Red Hat KVM solution is the same as that of the Dell PowerEdge R710 running the Microsoft Hyper-V solution.

Servers	Red Hat KVM solution with KSM	Microsoft Hyper-V solution
General dimension information		
Height (inches)	3.50	3.50
Width (inches)	17.50	17.50
Depth (inches)	27.00	27.00
U size in server rack (U)	2	2
Power supplies		
Total number	2	2
Wattage of each (W)	870	870
Cooling fans		
Total number	5	5
Dimensions (h x w) of each	2.5" x 2.5"	2.5" x 2.5"
Voltage (V)	12	12
Amps (A)	1.60	1.60
General processor setup		
Number of processor packages	2	2
Number of cores per processor package	4	4
Number of hardware threads per core	2	2
System power management policy	OS control	OS control
CPU		
Vendor	Intel	Intel
Name	Xeon X5550	Xeon X5550
Stepping	4	4
Socket type	LGA1366	LGA1366
Core frequency (GHz)	2.66	2.66
L1 cache	32 KB + 32 KB (per core)	32 KB + 32 KB (per core)
L2 cache	1 MB (4 x 256 KB)	1 MB (4 x 256 KB)
L3 cache	1 x 8 MB	1 x 8 MB
Platform		
Vendor and model number	Dell PowerEdge R710	Dell PowerEdge R710
Motherboard model number	PWBYN967	PWBYN967
Motherboard revision number	X08	X08
BIOS name and version	Dell 0.2.15	Dell 0.2.15

Servers	Red Hat KVM solution with KSM	Microsoft Hyper-V solution
BIOS settings	Virtualization enabled	Virtualization enabled
Memory modules		
Vendor and model number	Crucial CT51272BB1339	Crucial CT51272BB1339
Type	PC3-10600 DDR3	PC3-10600 DDR3
Speed (MHz)	1,333	1,333
Speed in the system currently running @ (MHz)	1,333	1,333
Timing/latency (tCL-tRCD-iRP-tRASmin)	9-9-9-24	9-9-9-24
Size (GB)	24	24
Number of RAM modules	6 x 4 GB	6 x 4 GB
Chip organization	Double-sided	Double-sided
Hard disk		
Vendor and model number	Seagate ST973451SS	Seagate ST973451SS
Number of disks in system	8	8
Size (GB)	73	73
Buffer size (MB)	16	16
RPM	15,000	15,000
Type	SAS	SAS
Operating system		
Name	Red Hat Enterprise Linux 5.4 beta	Windows Server 2008 R2
Build number	kernel-2.6.18-159.el5	7600
File system	ext3 (server) nfs (virtual machines)	nfs (server) nfs (virtual machines)
Language	English	English
Network card/subsystem		
Vendor and model number	Broadcom NetXtreme II BCM5709 Ethernet Controller x 4	Broadcom NetXtreme II BCM5709 Ethernet Controller x 4
Type	Integrated	Integrated
Optical drive		
Vendor and model number	TEAC DV-28S-VDB	TEAC DV-28S-VDB
USB ports		
Number	4	4
Type	2.0	2.0

Figure 4: Detailed system configuration information for the two test servers.

About Principled Technologies

We provide industry-leading technology assessment and fact-based marketing services. We bring to every assignment extensive experience with and expertise in all aspects of technology testing and analysis, from researching new technologies, to developing new methodologies, to testing with existing and new tools.

When the assessment is complete, we know how to present the results to a broad range of target audiences. We provide our clients with the materials they need, from market-focused data to use in their own collateral to custom sales aids, such as test reports, performance assessments, and white papers. Every document reflects the results of our trusted independent analysis.

We provide customized services that focus on our clients' individual requirements. Whether the technology involves hardware, software, Web sites, or services, we offer the experience, expertise, and tools to help you assess how it will fare against its competition, its performance, whether it's ready to go to market, and its quality and reliability.

Our founders, Mark L. Van Name and Bill Catchings, have worked together in technology assessment for over 20 years. As journalists, they published over a thousand articles on a wide array of technology subjects. They created and led the Ziff-Davis Benchmark Operation, which developed such industry-standard benchmarks as Ziff Davis Media's Winstone and WebBench. They founded and led eTesting Labs, and after the acquisition of that company by Lionbridge Technologies were the head and CTO of VeriTest.



Principled Technologies, Inc.
1007 Slater Rd., Suite 250
Durham, NC 27703
www.principledtechnologies.com
info@principledtechnologies.com

Principled Technologies is a registered trademark of Principled Technologies, Inc.
All other product names are the trademarks of their respective owners.

Disclaimer of Warranties; Limitation of Liability:

PRINCIPLED TECHNOLOGIES, INC. HAS MADE REASONABLE EFFORTS TO ENSURE THE ACCURACY AND VALIDITY OF ITS TESTING, HOWEVER, PRINCIPLED TECHNOLOGIES, INC. SPECIFICALLY DISCLAIMS ANY WARRANTY, EXPRESSED OR IMPLIED, RELATING TO THE TEST RESULTS AND ANALYSIS, THEIR ACCURACY, COMPLETENESS OR QUALITY, INCLUDING ANY IMPLIED WARRANTY OF FITNESS FOR ANY PARTICULAR PURPOSE. ALL PERSONS OR ENTITIES RELYING ON THE RESULTS OF ANY TESTING DO SO AT THEIR OWN RISK, AND AGREE THAT PRINCIPLED TECHNOLOGIES, INC., ITS EMPLOYEES AND ITS SUBCONTRACTORS SHALL HAVE NO LIABILITY WHATSOEVER FROM ANY CLAIM OF LOSS OR DAMAGE ON ACCOUNT OF ANY ALLEGED ERROR OR DEFECT IN ANY TESTING PROCEDURE OR RESULT.

IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC. BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH ITS TESTING, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC.'S LIABILITY, INCLUDING FOR DIRECT DAMAGES, EXCEED THE AMOUNTS PAID IN CONNECTION WITH PRINCIPLED TECHNOLOGIES, INC.'S TESTING. CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES ARE AS SET FORTH HEREIN.