

# Red Hat® Enterprise Virtualization 3.0



outperformed Microsoft® Hyper-V™ Server 2008 R2 on SPECjbb® 2005  
in eight Red Hat Enterprise Linux® 6.0 guests with 33% processor oversubscription

Because you want to maximize your server performance in a virtualized environment, you seek a hypervisor that can use your server's enhanced CPU and memory capabilities. Virtual machines (VMs), or guests, running CPU- and RAM-intensive applications should perform better on a server with modern processors with additional cores and memory.

To determine how effectively two hypervisor managers maximized server performance in a virtual environment, we tested the Java™ performance of guests using the following hypervisor products, each running on its respective operating system:

- Red Hat Enterprise Virtualization 3.0 on Red Hat Enterprise Linux 6.1 with kernel 2.6.32-214.el6.x86\_64
- Microsoft Hyper-V Server 2008 R2 on Microsoft Windows Server® 2008 R2

We tested the performance of eight guests, each with one Java Virtual Machine (JVM) instance, hosted on a Dell™ PowerEdge™ R710 server with two Intel® Xeon® processor 5670s. At eight guests, the hypervisors oversubscribed the server's processor resources.

When supporting eight Red Hat Enterprise Linux 6.0 guests at 33 percent oversubscription of processor resources, Red Hat Enterprise Virtualization 3.0 performed better than Microsoft Hyper-V Server 2008 R2.

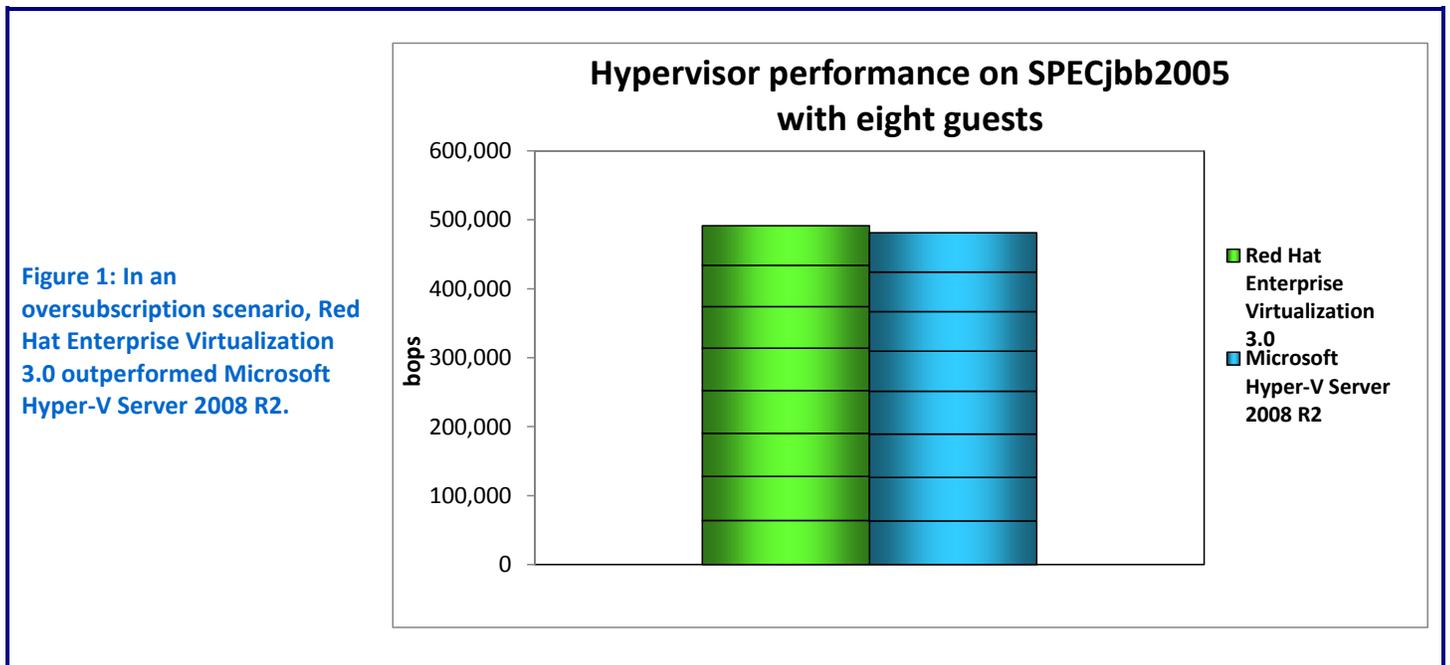
## MAXIMIZING SERVER PERFORMANCE IN A VIRTUAL ENVIRONMENT

The benefits of server virtualization were compelling enough that your company made the switch. While you have been enjoying savings in power, space, time, and money, you could be enjoying an even higher level of performance from your servers. The key to achieving optimal server performance in a virtualized environment is in selecting an effective hypervisor and operating system for the job.

We used the SPECjbb2005 benchmark with the OpenJDK 1.6.0\_17 JVM to test the two hypervisors, Red Hat Enterprise Virtualization 3.0 and Microsoft Hyper-V Server 2008 R2. The main SPECjbb2005 metric is business operations per second (bops). A higher SPECjbb2005 score indicates the server can handle more Java requests and thus deliver greater throughput. We explain the SPECjbb2005 benchmark in more detail in the [About SPECjbb2005 section](#).

Each result below is the median score of three benchmark runs. For complete details of the performance of each JVM by warehouse for each server, see the [What we found section](#).

As Figure 1 shows, with eight guests, Red Hat Enterprise Virtualization 3.0 achieved 491,352 bops, and Microsoft Hyper-V Server 2008 R2 achieved 481,633 bops.



The test design included NUMA and CPU affinity assignments of the hypervisors, where possible. The SPECjbb2005 benchmark is CPU and memory intensive, and these tunings can increase performance. Microsoft Hyper-V Server 2008 R2 does not support NUMA and CPU-affinity assignments for its guests. Red Hat Enterprise Virtualization, on the other hand, does support both of these tuning methods for its guests. We used the Red Hat Enterprise Virtualization Manager GUI to assign each guest its own NUMA policy and CPU affinities via the “numa” and “pincpu” hooks.<sup>1</sup> We also specified NUMA and CPU affinities for the SPECjbb2005 JVMs in the un-virtualized server tests.

## ABOUT SPECjbb2005

SPECjbb2005 is an industry-standard benchmark created by the Standard Performance Evaluation Corp. (SPEC) to measure a server’s Java performance. (Note: SPEC and the SPECjbb2005 are trademarks of the Standard Performance Evaluation Corporation.) SPEC modeled SPECjbb2005 on the three-tier client/server architecture, with the middle layer as the primary focus. According to SPEC, “Random input selection represents the first (user) tier. SPECjbb2005 fully implements the middle-tier business logic. The third tier is represented by tables of objects, implemented by Java Collections, rather than a separate database.”<sup>2</sup>

SPECjbb2005 utilizes multiple special data groups and multiple threads as it runs. Each data unit is a “warehouse,” a roughly 25MB collection of data objects. Each thread represents an active user posting transaction requests within a warehouse. The benchmark run begins with one warehouse and then increases the number of warehouses; its goal is to saturate the server’s processor capacity. As the number of warehouses increases, so does the number of threads. The benchmark’s metric portrays the server’s throughput in business operations per second or SPECjbb2005 bops. A higher number of SPECjbb2005 bops is better. (For more information on SPECjbb2005, go to [www.spec.org](http://www.spec.org).)

## WHAT WE FOUND

We examined the JVM performance for virtualized hosts having four virtual CPUs and 6 GB of memory. Figure 2 shows the median SPECjbb2005 results for the Red Hat Enterprise Virtualization 3.0 and Microsoft Hyper-V Server 2008 R2 hypervisors, respectively, for eight guests running simultaneously. SPECjbb2005 computes the score of each JVM by taking the average of the results during mixes when the server is running at peak performance. (In SPEC terms, these results are from “compliant” runs, which means we can disclose them publicly though we are not posting them on the SPEC Web

---

<sup>1</sup> For details on these Red Hat Enterprise Virtualization Manager guest-configuration hooks, see the Red Hat Enterprise Virtualization 3.0 Administration Guide, [http://docs.redhat.com/docs/en-US/Red\\_Hat\\_Enterprise\\_Virtualization/3.0/html/Administration\\_Guide/index.html](http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Virtualization/3.0/html/Administration_Guide/index.html)

<sup>2</sup> [www.spec.org/jbb2005/docs/UserGuide.htm](http://www.spec.org/jbb2005/docs/UserGuide.htm)

site with all the files SPEC requires. We include all the data necessary to reproduce these results.)

|                    | Eight guests                          |                                  |
|--------------------|---------------------------------------|----------------------------------|
|                    | Red Hat Enterprise Virtualization 3.0 | Microsoft Hyper-V Server 2008 R2 |
| Guest 1            | 64,126                                | 63,182                           |
| Guest 2            | 63,955                                | 63,172                           |
| Guest 3            | 62,457                                | 62,829                           |
| Guest 4            | 61,960                                | 62,422                           |
| Guest 5            | 61,303                                | 58,030                           |
| Guest 6            | 60,281                                | 57,443                           |
| Guest 7            | 60,152                                | 57,369                           |
| Guest 8            | 57,118                                | 56,825                           |
| <b>Total score</b> | <b>491,352</b>                        | <b>481,633</b>                   |

Figure 2: SPECjbb2005 results, in bops, for eight guests on each hypervisor. Higher numbers are better.

Next, to establish a baseline, we tested SPECjbb2005 performance on the host server with no virtualization. We used six independent JVM instances, each bound to two processor cores (four CPUs) to match the number of virtual CPUs per guest in the previous tests. The un-virtualized test running on Red Hat Enterprise Linux 6.0 achieved 507,475 bops. The performance decrease due to virtualization was small: 3 percent for Red Hat Enterprise Virtualization 3.0 and 5 percent for Microsoft Hyper-V Server 2008 R2.

## FINAL THOUGHTS

At eight guests, with the server's processor oversubscribed, Red Hat Enterprise Virtualization 3.0 demonstrated its ability to scale its workload to match the increasing demand, and did so better than Microsoft Hyper-V Server 2008 R2. This ability can help ensure excellent server performance in a virtualized environment, even when overcommitting resources.

## APPENDIX A – SERVER CONFIGURATION INFORMATION

Figure 3 provides detailed configuration information for the server we used in our testing.

| System                                | Dell PowerEdge R710                              |
|---------------------------------------|--|
| <b>Power supplies</b>                 |  |
| Total number                          | 1  |
| Vendor and model number               | Dell A570P-00                                    |
| Wattage of each (W)                   | 570  |
| <b>Cooling fans</b>                   |  |
| Total number                          | 5  |
| Dimensions (h x w) of each            | 2-1/2" x 2-1/2"                                  |
| Volts                                 | 12   |
| Amps                                  | 1.6  |
| <b>General</b>                        |  |
| Number of processor packages          | 2  |
| Number of cores per processor         | 6  |
| Number of hardware threads per core   | 2  |
| <b>CPU</b>                            |  |
| Vendor                                | Intel  |
| Name                                  | Xeon   |
| Model number                          | X5670  |
| Stepping                              | 01   |
| Socket type                           | LGA1366  |
| Core frequency (GHz)                  | 2.93   |
| Bus frequency (GT/s)                  | 6.40   |
| L1 cache (KB)                         | 32 + 32 (per core)                               |
| L2 cache (KB)                         | 256 (per core)                                   |
| L3 cache (MB)                         | 12   |
| <b>Platform</b>                       |  |
| Vendor and model number               | Dell PowerEdge R710                              |
| Motherboard model number              | PWB9YY69   |
| Motherboard chipset                   | Intel 5520                                       |
| BIOS name and version                 | Dell Incorporated 2.2.2 (9/21/2010) Revision 2.2 |
| BIOS settings                         | Default  |
| <b>Memory module(s)</b>               |  |
| Total RAM in system (GB)              | 48   |
| Vendor and model number               | Samsung M393B1K70BH1-CH9                         |
| Type                                  | PC3-10600R                                       |
| Speed (MHz)                           | 1,333  |
| Speed running in the system (MHz)     | 1,333  |
| Timing/Latency (tCL-tRCD-tRP-tRASmin) | 9-9-9-36   |
| Size (GB)                             | 8  |
| Number of RAM module(s)               | 6  |
| Chip organization                     | Double-sided                                     |
| Rank                                  | Dual   |

| System                               | Dell PowerEdge R710                                       |
|--------------------------------------|---|
| <b>Hard disk</b>                     |   |
| Vendor and model number              | FUJITSU MBB2147RC   |
| Number of disks in system            | 2   |
| Size (GB)                            | 146   |
| Buffer size (MB)                     | 16  |
| RPM                                  | 10,000  |
| Type                                 | SAS   |
| <b>Disk controller</b>               |   |
| Vendor and model                     | LSI™ Logic / Symbios Logic SAS1068E                       |
| Controller driver (module)           | 2.6.32-71.el6.x86_64                                      |
| Controller driver version            | 0B4D557979D0BC8F39D9984                                   |
| Controller firmware                  | 0.25.47.00-IR   |
| RAID configuration                   | RAID 1  |
| <b>Hypervisor 1 Operating system</b> |   |
| Name                                 | Red Hat Enterprise Linux 6.1                              |
| File system                          | ext4  |
| Kernel                               | 2.6.32-214.el6.x86_64                                     |
| Language                             | English   |
| Additional software                  | Red Hat Enterprise Virtualization 3.0                     |
| <b>Hypervisor 2 Operating system</b> |   |
| Name                                 | Microsoft Windows Server 2008 R2                          |
| File system                          | NTFS  |
| Language                             | English   |
| Version                              | 6.1 (Build 7601: Service Pack 1)                          |
| Hot Fixes                            | All publically available hot fixes as of 10 February 2011 |
| <b>Guest Operating system</b>        |   |
| Name                                 | Red Hat Enterprise Linux 6.0                              |
| File system                          | ext4  |
| Kernel                               | 2.6.32-71.el6.x86_64                                      |
| Language                             | English   |
| <b>Graphics</b>                      |   |
| Vendor and model number              | Matrox® G200eW  |
| <b>Ethernet</b>                      |   |
| Vendor and model number              | Broadcom® NetXtreme® II BCM57710                          |
| Type                                 | PCI Express   |
| Driver (Module)                      | 2.6.32-214.el6.x86_64                                     |
| Driver Version                       | 2.1.1   |
| <b>Optical drive(s)</b>              |   |
| Vendor and model number              | TEAC DV-28S   |
| Type                                 | DVD ROM   |
| <b>USB ports</b>                     |   |
| Number                               | 4   |
| Type                                 | 2.0   |

Figure 3: Detailed configuration information for the test server hosting the hypervisors.

## APPENDIX B - HOW WE TESTED

We performed SPECjbb2005 tests in three scenarios: un-virtualized Red Hat Enterprise Linux 6.0 server, eight Red Hat Enterprise Linux 6.0 guests on Red Hat Enterprise Virtualization, and eight Red Hat Enterprise Linux 6.0 guests on Microsoft Hyper-V Server 2008 R2. Tests on the un-virtualized server establish a base-line performance.

### Scenario 1: SPECjbb2005 JVM instances running on the un-virtualized server

The un-virtualized configuration is the Dell PowerEdge R710 server with no hypervisor, and the same OS as the guests. We ran six JVM instances with each allocated its own cpuset (2 cores or 4 CPUs). We also ran the SPECjbb collector process to gather results from the JVM instances. We chose six JVM instances in order to use every CPU on the server (fully subscribed).

#### Configuring the Dell PowerEdge R710 server

1. We configured the BIOS settings as we describe in the Common Configuration section below.
2. We created one RAID 1 volume from two 146GB disks.

#### Installing and configuring the OS and SPECjbb2005 on the un-virtualized server

1. Install Red Hat Enterprise Linux 6.0 on the Dell PowerEdge R710 according to the instructions in the Common Configuration section.
2. Log onto the system as `root`
3. Install the SPECjbb2005 software according to the instructions in the Common Configuration section.
4. Configure SPECjbb2005 for six JVM instances by editing the SPECjbb.prop file .

#### Running the SPECjbb2005 benchmark on the un-virtualized server

1. Execute the bash script `run_multiSPECjbb.sh`, given below, to perform the test.
2. The results are in the file `MultiVMReport.txt` in the directory created for this run in `/root/tests/SPECjbb2005_v1.07/results/SPECjbbMultiJVM.XXX`.

#### `run_multiSPECjbb.sh`

```
#!/bin/bash

# print JVM version information
cd /root/tests/SPECjbb2005_v1.07
date
export CLASSPATH=./jbb.jar:./check.jar:$CLASSPATH
echo JVM classpath = $CLASSPATH
java -fullversion

JVM=6

# JVM options
## memory:
jvm_opts1="-Xms5g -Xmx5g -Xmn4g"

## garbage-collection options:
jvm_opts2="-XX:+UseParallelOldGC -XX:ParallelGCThreads=8 -XX:+AlwaysPreTouch"

## other JVM optimizations
```

```

jvm_opts3="-XX:+AggressiveOpts -XX:+UseBiasedLocking -XX:+UseCompressedOops -
XX:AllocatePrefetchDistance=192 -XX:AllocatePrefetchLines=4 -XX:SurvivorRatio=68 -
XX:TargetSurvivorRatio=90 -XX:LoopUnrollLimit=45"

jvm_opts="$jvm_opts1 $jvm_opts2 $jvm_opts3"

#####

echo Starting Controller
java -cp jbb.jar:check.jar -Xms32m -Xmx32m spec.jbb.Controller \
    -propfile SPECjbb.props &
sleep 5

# each JVM is assigned 4 CPUs
let "cpu1=0"; let "cpu2=12"
let "cpu3=6"; let "cpu4=18"
for x in `seq $JVM`; do
    echo Starting instance $x on cpus: $cpu1, $cpu2, $cpu3, and $cpu4
    numactl -l -C $cpu1,$cpu2,$cpu3,$cpu4 java $jvm_opts spec.jbb.JBBmain \
        -propfile SPECjbb.props -id $x > multi_$$.$x &
    let "cpu1++"; let "cpu2++"
    let "cpu3++"; let "cpu4++"
done

## end of file

```

## Scenario 2: SPECjbb2005 JVM instances running in Red Hat Enterprise Linux 6.0 guests with Red Hat Enterprise Virtualization

We ran the SPECjbb2005 benchmark on Red Hat Enterprise Linux 6.0 guests under Red Hat Enterprise Virtualization 3.0 on Red Hat Enterprise Linux 6.1 with kernel 2.6.32-214.el6.x86\_64. We also used the Red Hat Enterprise Virtualization Manager to control the guests. To perform the test, we installed the Red Hat Enterprise Linux 6.1 OS with kernel 2.6.32-214.el6.x86\_64 on the Dell PowerEdge R710 server, installed Red Hat Enterprise Virtualization on the server using Red Hat Enterprise Virtualization Manager, and created eight Red Hat Enterprise Linux 6.0 guests with SPECjbb2005 installed.

### Configuring the Dell PowerEdge R710 server

1. We configured the BIOS settings as we describe in the Common Configuration section below.
2. We created one RAID1 volume from two 146GB disks.

### Installing and configuring Red Hat Enterprise Linux 6.1 with kernel 2.6.32-214.el6.x86\_6 on the host server

1. Insert and boot from the Red Hat Enterprise Linux 6.1 installation DVD.
2. Press Enter to install using graphical mode.
3. At the media test screen, select Skip, and click Next.
4. At the Red Hat Enterprise Linux 6.1 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 Devices screen, select Basic Storage Devices, and click Next.
8. Enter the system's hostname, and select Configure Network.
9. At the Configure Network Connections screen, select the network interface corresponding to the Red Hat Enterprise Virtualization Manager network, such as em0, and press Edit.

10. On the Connection Configuration screen, select the IPV4. Change Method to manual, and press Add. Enter the IP address, netmask prefix, and gateway. Enter the IP address of the DNS server. Press Apply.
11. At the Configure Network Connections screen, click Close.
12. At the Time zone selection screen, select the appropriate time zone, and click Next.
13. Enter the root password in the Root Password and Confirm fields, and click Next.
14. At the type of installation screen, select Use All Space, check Review and modify partitioning layout, and click Next.
15. At the partitioning layout screen, select lv\_home, and click Delete. At the Confirm Delete warning, click Delete.
16. Select lv\_root and click Edit. In the Size field, enter the Max size displayed. Click OK.
17. Click Next. If a warning appears, click Write changes to disk.
18. At the Boot loader screen, click Next.
19. At the default installation screen, select Basic Server, Customize now, and click Next.
20. Installation will begin automatically.
21. At the Congratulations screen, click Reboot.
22. After the system reboots and the first boot Welcome screen appears, click Forward.
23. Agree to the EULA, and click Forward.
24. At the Set Up Software Updates screen, select No, I prefer to register at a later time, and click Forward.
25. If a pop-up appears asking if you are sure, click No thanks.
26. At the Finish Updates Setup, click Forward.
27. At the Create User Screen, enter username `testuser` and password `Password1` and click Forward.
28. If a warning appears, click Yes.
29. At the Date and Time screen, enter the time and date, and select Synchronize time over the network.
30. Log into the server as `root` to complete the OS configuration.
31. Disable SELinux for these tests. Edit the file `/etc/selinux/config` and change the line `SELINUX=enforcing` to `SELINUX=disabled`
32. Install the 2.6.32-214.el6 kernel.  

```
# yum install kernel-2.6.32-214 kernel-firmware-2.6.32-214
```
33. Synchronize system time to the same source as the Red Hat Enterprise Virtualization Manager/Identity server. Edit `/etc/ntp.conf` to add the new time server, and run these following shell commands:  

```
# chkconfig ntpdate on
# service ntpdate start
```
34. Add the IP addresses for the host and Red Hat Enterprise Virtualization Manager servers to `/etc/hosts`.
35. Disable IPTABLES if possible or modify its configuration as below to allow incoming connections from 22/tcp for SSH, 16509/tcp for libvirt, and 54321 for vdsmd as a minimum.  

```
# chkconfig iptables off
```
36. Reboot the server with the following command:  

```
# shutdown -r now
```

## Installing and configuring the Red Hat Enterprise Virtualization on the host server from Red Hat Enterprise Virtualization Manager

1. Log onto the host server as `root`
2. Mount the RHEL 6.1 installation DVD at `/mnt` with the following command:  

```
# mount /dev/cdrom /mnt
```
3. Install the distribution GPG keys from DVD with the following command:  

```
# for i in /mnt/RPM-GPG-KEY-redhat-* ; do
> echo $i; rpm --import $i
> done
```
4. Copy or mount the Red Hat Enterprise Virtualization software to `/opt/RHEV`.
5. Copy this example yum repo file `rhev-install.repo` to `/etc/yum.repos.d/`

```

# rhel61-repo
[rhel61-local]
name=rhel61-local
baseurl=file:///mnt/Server
enabled=1
gpgcheck=1
skip_if_unavailable=1

[HA]
name=ha
baseurl=file:///mnt/HighAvailability
enabled=1
gpgcheck=1
skip_if_unavailable=1

[rhev-m]
name=latest rhvm
baseurl=file:///opt/RHEV/rhvm
enabled=1
gpgcheck=1
skip_if_unavailable=1

[rhev-agent]
name=latest rhvm agent
baseurl=file:///opt/RHEV/rhev-agent
enabled=1
gpgcheck=1
skip_if_unavailable=1

```

6. On the Red Hat Enterprise Virtualization Manager hosts tab, click New.
7. On the New Host screen, enter the name, IP address, and root password for the host server.
8. Ignore the Power Management Config warning screen as this test does not require server fencing. Click OK to continue to install.
9. Eject the DVD when you see the message Host rhel-01 installed. The host server is automatically rebooting.
10. After the host reboots, its name will appear on the Red Hat Enterprise Virtualization Manager hosts tab. Select the host, and click Activate.
11. On the virtualization host, install the NUMA and CPU affinity hooks using the following command:  

```
# yum install vdsmd-hook-numa vdsmd-hook-pincpu
```
12. From the Red Hat Enterprise Virtualization Manager hosts tab, put host into maintenance mode by selecting the host and clicking Maintenance.
13. On the virtualization host, restart the vdsmd daemon using the following command:  

```
# service vdsmd restart
```
14. Activate the host as in Step 10.

## Installing and configuring the Red Hat Enterprise Linux 6.0 guests from Red Hat Enterprise Virtualization Manager

1. Create the first Red Hat Enterprise Linux guest using Red Hat Enterprise Virtualization Manager. On the virtual machines tab, click New Server.
2. On the New Virtual Machine screen, select the General tab.
3. Enter the guest's name, 6 GB for Memory Size, and select four CPU sockets on the slider. Select Red Hat Enterprise Linux 6.X x64 for the operating system. Click OK.
4. On the New Virtual Machine – Guide Me screen, click Configure Virtual Disks.
5. On the New Virtual Disk screen, enter the disk size in GB, and click OK.

6. On the New Virtual Machine – Guide Me screen, click Configure Network Interfaces.
7. Click OK to accept the default properties.
8. Click Configure Later to close the New Virtual Machine – Guide Me screen.
9. Right-click the guest, and select Run Once.
10. On the screen, check Attach CD, and select the Red Hat Enterprise Linux 6.0 ISO image.
11. Click OK.
12. Open a console window.
13. Install the guest operating system following the instructions in Common Configuration for Installing and configuring Red Hat Enterprise Linux 6.0 x86\_64.
14. Log onto the guest, and install SPECjbb2005 following the instructions in Common Configuration for Installing and configuring SPECjbb below.
15. Install the SPECjbb2005 software on the first guest according to the instructions in the Common Configuration section.
16. Configure SPECjbb for one JVM instance by editing the SPECjbb.prop file .
17. Copy the bash script runSPECjbb.sh, given below, into root's home directory /root.
18. Shutdown the guest.
19. Create a guest template of this first guest from the Red Hat Enterprise Virtualization Manager virtual machines tab.
20. Create a new guest from this template from the Red Hat Enterprise Virtualization Manager virtual machines tab. The new disk should be pre-allocated.
21. Start the guest, open a console window, and log in as `root`
22. Update the new guest's hostname by editing the file `/etc/sysconfig/network` and changing the line beginning with `HOSTNAME=`.
23. Update the guest's IP address by editing the file `/etc/sysconfig/network-scripts/ifcfg-eth0` and changing the line beginning with `IPADDR=`. Delete the line beginning with `HWADDR=` if it exists.
24. Reset the udev network-device mapping by editing the file `/etc/udev/rules.d/70-persistent-net.rules`, and deleting its contents. Save the empty file but do not delete it.
25. Repeat steps 9 through 14 for the remaining guests.

### Running the SPECjbb2005 benchmark on the Red Hat Enterprise Virtualization Manager guests

1. Start the eight guests from Red Hat Enterprise Virtualization Manager virtual machines tab.
2. Log onto the host server as `root`
3. Start the SPECjbb2005 benchmark by running the bash script `runSPECjbbGuests.sh`
4. Collect the SPECjbb2005 BOPs from each guest by running the bash script `getSPECjbbGuests.sh`. Note that you must specify the run number on the command line.

#### `runSPECjbbGuests.sh`

```
#!/bin/bash

## Note: we assume the guests' IP Addresses are sequential in the form 10.41.5.XXX
date
for i in `seq 141 148`; do
    echo $i
    ssh 10.41.5.$i 'nohup ./runSPECjbb.sh 2>&1 > out; echo -n "\"$i' "; date' &
done

## end of file
```

#### `getSPECjbbGuests.sh`

```
#!/bin/bash
```

```

## argument #1:
## the SPECjbb run as a three-digit number; e.g., 001, ...,010, ...

# location of the SPECjbb results on each guest
r_file=tests/SPECjbb2005_v1.07/results/SPECjbbSingleJVM/SPECjbb.$1.txt

## Note: we assume the guests' IP Addresses are sequential in the form 10.41.5.XXX
for i in `seq 141 148` ; do
    echo -n "$i ";
    ssh 10.41.5.$i "head -1 $r_file"
done

## end of file

```

### **runSPECjbb.sh**

```

#!/bin/bash

## install this script on each guest

# print JVM version information
cd ~/SPECjbb2005_v1.07
date
export CLASSPATH=./jbb.jar:./check.jar:$CLASSPATH
echo $CLASSPATH
java -fullversion

# JVM options
## memory:
jvm_opts1="-Xms5g -Xmx5g -Xmn4g"

## garbage-collection options:
jvm_opts2="-XX:+UseParallelOldGC -XX:ParallelGCThreads=8 -XX:+AlwaysPreTouch"

## other JVM optimizations
jvm_opts3="-XX:+AggressiveOpts -XX:+UseBiasedLocking -XX:+UseCompressedOops -XX:AllocateP
refetchDistance=192 -XX:AllocatePrefetchLines=4 -XX:SurvivorRatio=68 -XX:TargetSurvivorRa
tio=90 -XX:LoopUnrollLimit=45"

# run one instance using parameters from the file SPECjbb.props
java $jvm_opts1 $jvm_opts2 $jvm_opts3 spec.jbb.JBBmain -propfile SPECjbb.props

## end of file

```

## **Scenario 3: SPECjbb2005 JVM instances running in Red Hat Enterprise Linux 6.0 guests with Microsoft Hyper-V Server**

We ran the SPECjbb2005 benchmark on Red Hat Enterprise Linux 6.0 guests under Microsoft Hyper-V Server 2008 R2. To do so, we installed Microsoft Windows Server on the Dell PowerEdge R710 server, configured Hyper-V, and created eight Red Hat Enterprise Linux 6.0 guests with SPECjbb2005 installed.

### **Configuring the Dell PowerEdge R710 server**

1. We configured the BIOS settings we describe in the Common Configuration section below.
2. We created one RAID1 volume from two 146GB disks.

## Installing Windows Server 2008 R2 and Hyper-V

1. Insert and boot from the Microsoft Windows Server 2008 R2 installation DVD.
2. At the Install Windows screen, select English, English (United States), and US for the installation language, time format, and keyboard. Click Next.
3. On the next screen, click the arrow.
4. On the Select operating system screen, select Windows Server 2008 R2 Enterprise (Full Installation), and click Next.
5. At the Microsoft Software License screen, select I accept the license terms, and click Next.
6. Select Custom Install.
7. At the Where do you want to install Windows screen, select the correct disk, and click Next to start the installation process. It will reboot in the middle of the installation and continue the installation.
8. At the change password screen, click OK, enter the Administrator's password twice, and click the arrow.
9. At Your password has been changed screen, click OK to log on.
10. Windows will start a wizard for completing Initial Configuration tasks.
11. Click Set time zone in order to set the date, time and time zone as well as to configure automatic DST switching, and SNTP servers.
12. Click Configure networking to open the Network Connections window.
13. Right-click the appropriate network adapter, and select Properties from the pop-up menu.
14. At the Networking screen, select Internet Protocol Version 4 (TCP/IPv4), and click Properties.
15. At the Internet Protocol Version 4 (TCP/IPv4) Properties window, select Use the following IP address, and enter the IP Address, Subnet mask, and Default gateway (if needed).
16. Select Use the following DNS server address, and enter the Preferred and Alternate DNS server IP addresses.
17. Click OK to return to Networking. Click OK to return to the Network Connections window. Press Ctrl-W to close and return to the setup wizard.
18. Click Enable Remote Desktop.
19. At the Remote tab, select Allow connections from computers running any version of Remote Desktop Protocol. Click OK on the Firewall exception pop-up, and click OK to finish configuring remote desktop.
20. Click Add Role to enable Hyper-V.
21. At the Before You Begin screen, click Next.
22. At the Server Roles screen, select Hyper-V, and click Install.
23. Restart the server.

## Creating the Red Hat Enterprise Linux 6.0 guests in Hyper-V

1. Log onto the Windows systems as `administrator`
2. Open the Hyper-V Manager.
3. Right-click the Hyper-V hostname, and select New→Virtual Machine.
4. At the Before you Begin screen, click Next.
5. At the Specify Name and Location screen, enter the guest's name, select a logical unit number (LUN) for the OS, and click Next.
6. At the Assign Memory screen, type `6144 MB` and click Next.
7. At the Configure Networking screen, do not enter a network adapter (see below), and click Next.
8. At the Connect Virtual Hard Disk screen, select Create a new virtual hard disk, enter a name and location for the VHD file, set the disk size to 16 GB, and click Next.
9. At the Installation Options screen, select Install an operating system from a boot CD/DVD-ROM, select Image file (.iso), and browse to the Red Hat Enterprise Linux 6.0 x86\_64 installation DVD file. Click Finish.
10. Click Finish to create the guest.
11. In the Virtual Machines section of Hyper-V Manager, right-click the guest, and click Settings.
12. Click Processor and change your number of virtual processors to 4.

13. Click Add Hardware, select Legacy Network Adapter, and click Add.
14. On the Network Adapter screen, select the SPECjbb client network from the pull-down menu, and click OK to complete the configuration.
15. Start the guest so that it boots from the ISO image.
16. Install the guest operating system following the instructions in Common Configuration for Installing and configuring Red Hat Enterprise Linux 6.0 x86\_64.
17. Log onto the guest, and install SPECjbb2005 following the instructions in Common Configuration for Installing and configuring SPECjbb below.
18. Shutdown the guest, and clone this first guest seven times.
19. Start the first clone, and log onto it.
20. Change the guest's name by editing the file `/etc/sysconfig/network`, and change the value of `HOSTNAME`
21. Change the guest's IP address by editing the file `/etc/sysconfig/network-scripts/ifcfg-eth0` to change the value of `IPADDR`. Remove the line starting with `HWADDR=` if it exists.
22. Delete all lines from the file `/etc/udev/rules.d/70-persistent-net.rules`.
23. Reboot the guest.
24. Repeat steps 19 through 23 for the remaining cloned guests.

### Running the SPECjbb benchmark on the Hyper-V managed guests

1. Reboot the Windows server.
2. Log onto the server as `administrator`
3. Open the Hyper-V Manager.
4. Start one Linux guest.
5. Open a console window on one guest and log in as `root`
6. At the prompt, type `bash runSPECjbb.sh` but do not press Enter.
7. Repeat steps 4 through 6 for each guest.
8. To start the run, press Enter in each console window.

### Common configurations

Several sets of tasks are common to the three test configurations. We describe them here.

#### Configuring the Dell PowerEdge R710 Server for virtualization

1. Turn on the server, and press F12 to enter the BIOS menu.
2. In the processor section, we enabled these settings: all CPUs, Hyperthreading, VTd, QPI compute, Prefetch, DCU Stream, and Data Reuse.
3. In the performance section, we enabled C-States, C1E, and Turbo mode. We also set the Power Option to OS Control.
4. Exit the BIOS menu, saving the changes.

#### Installing and configuring Red Hat Enterprise Linux 6.0 x86\_64

1. Insert and boot from the Red Hat Enterprise Linux 6.0 installation DVD.
2. Press Enter to install using graphical mode.
3. At the media test screen, select Skip, and click Next.
4. At the Red Hat Enterprise Linux 6.0 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 Devices screen, select Basic Storage Devices, and click Next.
8. Enter the system's hostname, and click Next.
9. At the Time zone selection screen, select the appropriate time zone, and click Next.
10. Enter the root password in the Root Password and Confirm fields, and click Next.

11. At the type of installation screen, select Use All Space, check Review and modify partitioning layout, and click Next.
12. At the partitioning layout screen, select lv\_home, and click Delete. At the Confirm Delete warning, click Delete.
13. Select lv\_root, and click Edit. In the Size field, enter the Max size displayed. Click OK.
14. Click Next. If a warning appears, click Write changes to disk.
15. At the Boot loader screen, click Next.
16. At the default installation screen, select Basic Server, Customize now, and click Next.
17. Installation will begin automatically.
18. At the Congratulations screen, click Reboot.
19. After the system reboots and the firstboot Welcome screen appears, click Forward.
20. Agree to the EULA, and click Forward.
21. At the Set Up Software Updates screen, select No, I prefer to register at a later time, and click Forward.
22. If a pop-up appears asking if you are sure, click No thanks.
23. At the Finish Updates Setup, click Forward.
24. At the Create User Screen, enter username `testuser` and password `Password1` and click Forward.
25. If a warning appears, click Yes.
26. At the Date and Time screen, enter the time and date, and select Synchronize time over the network.
27. Replace the default NTP servers with the local ones, and click Finish.
28. At the Kdump screen, uncheck Enable Kdump, and click Finish.

## Completing the initial Red Hat Enterprise Linux 6.0 configuration

Log into the server as `root` to complete the OS configuration and install third-party software by performing the following steps:

1. Disable SELinux for these tests because we have found that passwordless SSH can fail under the default SELinux configuration. Edit the file `/etc/selinux/config` and change the line `SELINUX=enforcing` to `SELINUX=disabled`
2. Disable these unused daemons with the following script:

```
# for s in auditd avahi-daemon bluetooth dnsmasq fcoe firstboot \
  ip6tables iptables irqbalance iscsi iscsid libvirt-guests lldpad \
  mdmonitor netconsole netfs nfs nfslock ntpdate postfix restorecond \
  rpcbind rpcgssd rpcidmapd rpcsvcgssd; do
    chkconfig $s off
    service $s stop
done
```

3. Ensure these services are enabled using the following script:

```
# for s in cpuspeed sshd sysstat; do
    chkconfig $s on
    service $s start
done
```

4. Configure the host's main IP address on interface `eth0`.
5. Replace the contents of `/etc/sysconfig/network-scripts/ifcfg-eth0` with the following script, where for purpose of illustration we use an IP address/netmask of `10.41.1.200/24`:

```
DEVICE="eth0"
ONBOOT="yes"
TYPE=Ethernet
BOOTPROTO=none
IPV4_FAILURE_FATAL=yes
```

```
IPV6INIT=no
IPADDR=10.41.1.200
PREFIX=24
```

6. Type the following command to restart networking to effect these changes:

```
# service network restart
```

7. Install SSH public/private keys. If this is the first system, generate the common key pair:

```
# ssh-keygen -t dsa -N "" -f ~/.ssh/id_dsa
```

For all other systems, copy the ssh key pair from the first system to the current one:

```
# scp -r first-system:~/.ssh ~
```

8. Reboot the system:

```
# shutdown -r now
```

### Installing and configuring SPECjbb2005 benchmark on Red Hat Enterprise Linux 6.0 systems

We obtained SPECjbb2005 version 1.07, dated March 15, 2006. We designed the present benchmark following the SPEC run rules.

1. Install SPECjbb2005 by copying the contents of the SPECjbb2005 CD to the directory `/root/tests/SPECjbb2005_v1.07`.
2. For the purposes of this project, the default settings in the SPECjbb properties file sufficed: we used one JVM instance with a maximum of eight warehouses on each guest
3. SPECjbb2005 requires a Java Virtual Machine on the system under test. We chose the one provided by Red Hat Enterprise Linux 6.0, OpenJDK 1.6.0\_17 (build 14.0-b16):  

```
# yum install java-1.6.0-openjdk-1.6.0.0\*
```

## ABOUT PRINCIPLED TECHNOLOGIES



Principled Technologies, Inc.  
1007 Slater Road, Suite 300  
Durham, NC, 27703  
[www.principledtechnologies.com](http://www.principledtechnologies.com)

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 our clients assess how it will fare against its competition, its performance, its market readiness, 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 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.

---