COMPARING JAVA PERFORMANCE: RED HAT ENTERPRISE LINUX 6 AND OPENJDK VS. MICROSOFT WINDOWS SERVER 2012 AND ORACLE JAVA HOTSPOT

Red Hat® Enterprise Linux® 6 and OpenJDK

delivered better Java® performance
on the SPECjbb2013® benchmark

versus Microsoft® Windows® Server 2012
with Oracle® Java Hotspot

redhat

Java language comes with an attractive promise of "write once, run anywhere," implying that the application code needs to be compiled only once. While "write once" is largely true, the challenges associated with "run anywhere" often leave users wondering what other factors they should be considering while selecting their next Java application platform.

Java Virtual Machine (JVM) provides a sufficiently rich abstraction layer to run applications independent of particular computer hardware implementation. However, JVMs typically do not run directly on hardware platforms and require an operating system (OS) to interact with the underlying hardware resources. That makes selection of the OS, along with hardware architecture and JVM implementation, an important step in defining any Java application platform.

To help you make this assessment, we used the SPECjbb2013[1] benchmark in the Principled Technologies labs, to compare the Java performance of two popular operating system and JVM implementations: Red Hat Enterprise Linux 6 with OpenJDK[2] (Red Hat/OpenJDK solution) and Microsoft Windows Server 2012 with Oracle Java HotSpot (Microsoft/Java HotSpot solution). To minimize differences in the hardware configuration and setup, we used the same server hardware to perform both tests.

We found that Red Hat Enterprise Linux 6 with OpenJDK outperformed Windows Server 2012 with Java Hotspot, achieving up to 1.7 percent more critical-jOPS

---

and 3.7 percent more max-jOPS using small heap settings.[3] Moreover, using a large heap size, the Red Hat/OpenJDK solution scored 22,126 critical-jOPS and 34,129 max-jOPS, while Windows Server 2012 and the Java HotSpot JVM did not report a qualifying result on the SPECjbb2013 benchmark when using a large heap.[4]

BETTER SPECJBB2013 PERFORMANCE

The operating system and the JVM that are being deployed on your servers can directly affect the performance your applications deliver. For our comparison of Red Hat Enterprise Linux 6 with OpenJDK and Microsoft Windows Server 2012 with Java HotSpot, we used the SPECjbb2013 benchmark with both large and small heap settings. For detailed system configuration information, see Appendix A. See Appendix B for step-by-step testing details.

Figure 1 shows the scores that the systems achieved on the SPECjbb2013 benchmark with both large heap and small Java heap settings. When the Java program starts, the JVM requests a memory block from the OS and allocates it to that application. That block of memory is consecutively called Java heap and is managed by JVM.

We used a 240GB heap size for the Red Hat/OpenJDK solution in the large heap scenario and a 20GB heap size in the small heap scenario. When using a 240GB heap size with the Microsoft/Java HotSpot solution, the benchmark failed to run. We then used a similar large heap (232GB) configuration, and the benchmark ran but did not report a qualifying score.

---

[3] Small heap (20GB) OpenJDK and Java HotSpot JVM results were not submitted to SPEC for publication.
[4] This result is an officially submitted SPECjbb2013 result at spec.org. See http://www.spec.org/jbb2013/results/res2013q2/jbb2013-20130319-00013.html for more details.

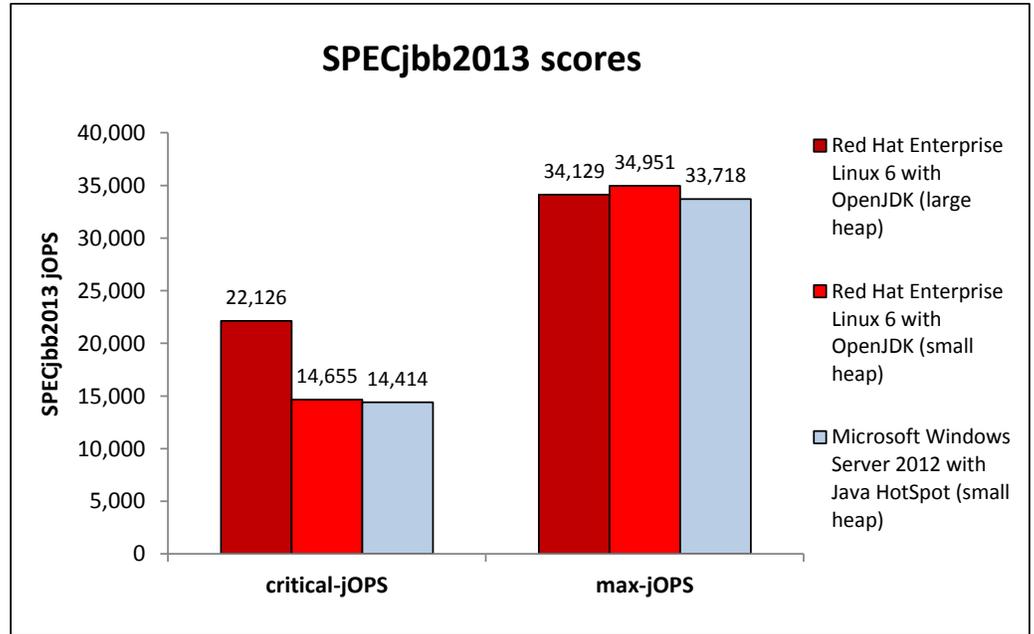**Figure 1: SPECjbb2013 results, in critical-jOPS and max-jOPS, for the solutions we tested. Higher numbers are better.**

Figure 2 details the SPECjbb2013 results for the solutions we tested.

	critical-jOPS	max-jOPS
Small heap		
Red Hat Enterprise Linux 6 with OpenJDK	14,655	34,951
Microsoft Windows Server 2012 with Java HotSpot	14,414	33,718
Red Hat Advantage	1.7%	3.7%
Large heap		
Red Hat Enterprise Linux 6with OpenJDK	22,126	34,129
Microsoft Windows Server 2012 with Java HotSpot	N/A	N/A

Figure 2: Large and small heap SPECjbb2013 results for the solutions.

WHAT WE TESTED
About Red Hat Enterprise Linux 6

Red Hat Inc. positions Red Hat Enterprise Linux 6 as a prime competitor to proprietary operating systems found in enterprise data centers. Red Hat Enterprise Linux 6 is designed to deliver performance and scalability for both small and large servers with documented scalability up to 4,096 CPUs and 64 terabytes of RAM. It provides native support for the majority of the latest and most important enterprise data center technologies, such as 40Gb Ethernet networking and KVM virtualization as well as InfiniBand®, FCoE, and iSCSI protocols. According to Red Hat, Red Hat and its hardware partners are enabling reliability, availability, serviceability (RAS), and scalability features to help minimize downtime, increase availability, and protect data.

Red Hat includes as part of its Linux offering tested, open source applications. For more information about Red Hat Enterprise Linux 6, see
http://www.redhat.com/f/pdf/rhel/RHEL6_datasheet.pdf.

About SPECjbb2013

As SPEC describes at spec.org, "The SPECjbb2013 benchmark has been developed from the ground up to measure performance based on the latest Java application features. It is relevant to all audiences who are interested in Java server performance, including JVM vendors, hardware developers, Java application developers, researchers and members of the academic community."

As SPEC lists on its Web site, new features of SPECjbb2013 include:

- A usage model based on a worldwide supermarket company with an IT infrastructure that handles a mix of point-of-sale requests, online purchases, and data-mining operations.
- Both a pure throughput metric and a metric that measures critical throughput under service level agreements (SLAs) specifying response times ranging from 10ms to 500ms.
- Support for multiple run configurations, enabling users to analyze and overcome bottlenecks at multiple layers of the system stack, including hardware, OS, JVM, and application layers.
- Exercising new Java 7 features and other important performance elements, including the latest data formats (XML), communication using compression, and messaging with security.
- Support for virtualization and cloud environments.

For more information, visit http://www.spec.org/jbb2013/.

IN CONCLUSION

Both the operating system and JVM that you choose could affect the performance of your Java applications, so it is important that you choose a solution where each component delivers the best possible performance. As our results indicate, Red Hat Enterprise Linux 6 with OpenJDK outperformed Microsoft Windows Server 2012 with Java HotSpot on the industry-standard SPECjbb2013 benchmark on both of the reported metrics, max-jOPS and critical-jOPS, using small heap size. With large heap size, the Red Hat/OpenJDK solution delivered 34,129 max-jOPS and 22,126 critical-jOPS, the best reported critical operations score as of June 30, 2013, while Microsoft/Java HotSpot solution could not produce a qualifying benchmark result.

APPENDIX A – SYSTEM CONFIGURATION INFORMATION

Figure 3 provides detailed configuration information for the test systems.

System	Dell PowerEdge R720
Power supplies	
Total number	2
Vendor and model number	Dell D750E-S1
Wattage of each (W)	750
Cooling fans	
Total number	6
Vendor and model number	San Ace 60 9GA0612P1K641
Dimensions (h x w) of each	2-1/2" x 2-1/2"
Volts	12
Amps	0.95
General	
Number of processor packages	2
Number of cores per processor	8
Number of hardware threads per core	2
System power management policy	Performance Optimized
CPU	
Vendor	Intel®
Name	Xeon®
Model number	E5-2680
Stepping	C2
Socket type	FCLGA2011
Core frequency (GHz)	2.7
Bus frequency	8.00 GT/s
L1 cache	32 KB I + 32 KB D on chip per core
L2 cache	256 KB I+D on chip per core
L3 cache	20 MB I+D on chip per chip
Platform	
Vendor and model number	Dell PowerEdge R720
Motherboard model number	0M1GCR
BIOS name and version	1.5.1
BIOS settings	Performance Optimized power profile
Memory module(s)	
Total RAM in system (GB)	256
Vendor and model number	Samsung M393B2G70BH0-YH9
Type	PC3-10600R
Speed (MHz)	1,333
Speed running in the system (MHz)	1,333
Size (GB)	16
Number of RAM module(s)	16
Chip organization	Double-sided

System	Dell PowerEdge R720
Rank	Dual
Operating system # 1	
Name	Red Hat Enterprise Linux 6.4
Build number (kernel)	2.6.32-358.0.1.el6.x86_64
File system	ext4
Language	English
Operating system # 2	
Name	Microsoft Windows Server 2012
Build number (kernel)	9200
File system	NTFS
Language	English
Graphics	
Vendor and model number	Matrox® G200e
Graphics memory (MB)	8
RAID controller	
Vendor and model number	Dell PERC H710P Mini
Firmware version	21.1.0-0007
Driver version	LSI 5.2.122.0 (04/03/2012)
Cache size (MB)	1,024
Hard drives	
Vendor and model number	Intel SSDSA2BZ100G3
Number of drives	2
Size (GB)	100
RPM	N/A
Type	SSD
Ethernet adapters	
Vendor and model number	Intel I350 Quad Port Gigabit Network Adapter
Type	Integrated
Driver	Microsoft 12.0.150.0 (02/29/2012)
Optical drive(s)	
Vendor and model number	TEAC DV-28SW
Type	Integrated
USB ports	
Number	4 external, 1 internal
Type	2.0

Figure 3: Configuration information for our test system.

APPENDIX B - HOW WE TESTED

Installing the Red Hat Enterprise Linux 6.4 operating system

1. Insert and boot from the Red Hat Enterprise Linux 6.4 x86_64 installation DVD.
2. At the welcome screen, select Install or upgrade an existing system, and press Enter.
3. At the Media test screen, select Skip, and press Enter.
4. At the Red Hat Enterprise Linux 6 title screen, click Next.
5. At the Choose an Installation Language screen, select English, and click Next.
6. At the Keyboard Type screen, select U.S. English, and click Next.
7. At the Storage Devices screen, select Basic Storage Devices, and click Next.
8. If a warning for device initialization appears, select Yes, discard any data.
9. At the Name the Computer screen, type the host name, and click Configure Network.
10. At the Network Connections screen, select the server's main or management network interface, and click Edit.
11. At the Editing network interface screen, check Connect Automatically.
12. On the same screen, select the IPv4 Settings tab, change the Method to Manual, and click Add.
13. On the same screen, enter the IP address, Netmask, Gateway, and DNS server. Click Apply.
14. Click Close on the Network Connections screen, and click Next on the Name the Computer screen.
15. At the Time zone selection screen, select the appropriate time zone, and click Next.
16. Enter the root password in the Root Password and Confirm fields, and click Next.
17. At the Assign Storage Devices screen, from the list in the left column, select the Linux disk, and click the arrow to copy the device to the right column. Next to the Linux disk, click the Boot radio button, and click Next.
18. At the Partition selection screen, select Replace Existing Linux System(s), and click Next.
19. If a warning appears, click Write changes to disk.
20. At the default installation screen, click Next to begin the installation.
21. At the Congratulations screen, click Reboot.
22. After the system reboots, log in as root.
23. Ensure your system is updated via RHN
24. Download the SPECjbb2013 tar file from spec.org and extract on the system.

Installing Microsoft Windows Server 2012 Datacenter

1. Connect the installation media and restart the server.
2. When the option appears, press F11 to enter the Boot Manager.
3. Select SATA Optical Drive and press Enter.
4. Press any key when prompted to boot from DVD.
5. When the installation screen appears, click My language is English.
6. Leave language, time/currency format and input method as default, and click Next.
7. Click Install now.
8. When the Windows Setup window appears, click No thanks when prompted to go online to install updates.

9. Select Windows Server 2012 Datacenter (Server with a GUI), and click Next.
10. Check I accept the license terms, and click Next.
11. Click Custom: Install Windows only (advanced).
12. Press Alt+A to open advanced partition options. Delete any partitions until there is only Drive 0 Unallocated Space.
13. Select Drive 0 Unallocated Space, and click Next, at which point Windows will begin installing, and will restart automatically after completing.
14. When the Settings page appears, fill in the Password and Reenter Password fields with the same password.
15. Log in with the password you set up.
16. Download the SPECjbb2013 zip file from spec.org and extract on the system.

Configuring Windows Update
1. In the left pane of the Server Manager window, click Local Server.
2. In the main frame, next to Windows Update, click Not configured.
3. In the Windows Update window, in the main pane, click Let me choose my settings.
4. Under Important updates, select Never check for updates (not recommended), and then click OK.
5. In the left pane, click Check for updates, and install all available updates.
6. Close the Windows Update window.

Configuring Windows Firewall
1. In Server Manager, click Tools→Windows Firewall with Advanced Security.
2. In the Overview section, click Windows Firewall Properties.
3. In the Domain Profile tab, for Firewall state, click Off.
4. In the Private Profile tab, for Firewall state, click Off.
5. In the Public Profile tab, for Firewall state, click Off.
6. Click OK.
7. Close the Windows Firewall Properties window.

Installing the JVMs

Installing OpenJDK (Red Hat Enterprise Linux 6.4)
1. Install the OpenJDK packages using yum:
```
yum –y install openjdk-1.7.0
```

Installing Java HotSpot (Windows Server 2012)
1. Download the Oracle Java SE Runtime Environment for Windows x64 from the Oracle Web site.[5] We used the jre-7u15-windows-x64.exe version.
2. Run the downloaded installer and accept all default settings, except specify the installation folder as the following:
```
C:\jre-7u15
```

---

[5] http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html

Tuning the system and running the benchmark

We used the following bash script to execute the benchmark and relevant tuning parameters for each test. For each Java platform, we used JAVA and JAVA_OPTS variables as necessary by commenting out those not used. The script below captures all settings used during our testing.

Script for Red Hat Enterprise Linux 6.4

```bash
#!/bin/bash

################################################################
## Sample script for running SPECjbb2013 in Composite mode.
##
## This sample script demonstrates launching the Controller, TxInjector and
## Backend in a single JVM.
################################################################

# SYSTEM TUINIG
umount /mnt/libhugetlbfs
sync
sync
echo 0 > /proc/sys/vm/nr_hugepages
echo 3 > /proc/sys/vm/drop_caches
cat /proc/meminfo | grep Huge
echo 23000 > /proc/sys/vm/nr_hugepages
#echo 12300 > /proc/sys/vm/nr_hugepages
#echo 2000 > /proc/sys/vm/nr_hugepages
cat /proc/meminfo | grep Huge
mount -t hugetlbfs hugetlbfs /mnt/libhugetlbfs

setenforce 0

tuned-adm profile latency-performance
#echo always > /sys/kernel/mm/redhat_transparent_hugepage/enabled
cat /sys/kernel/mm/redhat_transparent_hugepage/enabled

sleep 5

# Benchmark options
SPEC_OPTS=""

# Java options for Composite JVM

# BEST FOR OpenJDK:
#JAVA_OPTS="-Xmx240g -Xms240g -Xmn220g -XX:+UseParallelOldGC -XX:+AggressiveOpts
-XX:ThreadPriorityPolicy=1 -XX:-UseBiasedLocking -
XX:+UseLargePagesIndividualAllocation"

# SMALL HEAP SETTINGS
JAVA_OPTS="-Xmx20G -Xms20G -Xmn15G -XX:+UseLargePages -XX:-UseBiasedLocking -
XX:+UseParallelOldGC"
```

```
################################################################
# This benchmark requires a JDK7 compliant Java VM.  If such a JVM is not on
# your path already you must set the JAVA environment variable to point to
# where the 'java' executable can be found.
################################################################

# OpenJDK
#JAVA=/usr/lib/jvm/jre-1.7.0-openjdk.x86_64/bin/java

which $JAVA > /dev/null 2>&1
if [ $? -ne 0 ]; then
    echo "ERROR: Could not find a 'java' executable. Please set the JAVA
environment variable or update the PATH."
    exit 1
fi

echo "Launching SPECjbb2013 in Composite mode..."
echo

echo "Start Composite JVM"
$JAVA $JAVA_OPTS $SPEC_OPTS -jar specjbb2013.jar -m COMPOSITE 2>composite.log >
composite.out &

    COMPOSITE_PID=$!
    echo "Composite JVM PID = $COMPOSITE_PID"

sleep 5

echo
echo "SPECjbb2013 is running..."

wait $COMPOSITE_PID
echo "Composite JVM has stopped"
echo

echo "SPECjbb2013 has finished"
echo

exit 0
```

Script for Windows Server 2012

```
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Sample script for running SPECjbb2013 in Composite mode.
::
:: This sample script demonstrates launching the Controller, TxInjector and
:: Backend in a single JVM.
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

@echo off

:: Benchmark options
set SPEC_OPTS=

:: Java options for Composite JVM

:: LARGE HEAP
REM set JAVA_OPTS=-Xmx232G -Xms232G -Xmn216G -XX:+UseParallelOldGC -XX:-
UseBiasedLocking -XX:+UseLargePages

:: SMALL HEAP
set JAVA_OPTS=-Xmx20G -Xms20G -Xmn15G -XX:+UseLargePages -XX:-UseBiasedLocking -
XX:+UseParallelOldGC

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: This benchmark requires a JDK7 compliant Java VM.  If such a JVM is not on
:: your path already you must set the JAVA environment variable to point to
:: where the 'java' executable can be found.
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

@set JAVA="C:\jre-7u15\bin\java.exe"

@set JAVAPATH=
@for %%J in (%JAVA%) do (@set JAVAPATH=%%~$PATH:J)

@if not defined JAVAPATH (
   echo ERROR: Could not find a 'java' executable. Please set the JAVA
environment variable or update the PATH.
   exit /b 1
) else (
   @set JAVA="%JAVAPATH%"
)

%JAVA% -version

echo Launching SPECjbb2013 in Composite mode...
echo.

echo Start Composite JVM
@echo on
```

```
%JAVA% %SPEC_OPTS% %JAVA_OPTS% -jar specjbb2013.jar -m COMPOSITE 2>composite.log
> composite.out
@echo off

echo SPECjbb2013 is running...
echo.

echo Composite JVM has stopped
echo.

echo SPECjbb2013 has finished
echo.

exit /b 0
```

ABOUT PRINCIPLED TECHNOLOGIES

Principled Technologies®

Principled Technologies, Inc.
1007 Slater Road, Suite 300
Durham, NC, 27703
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.