PERFORMANCE COMPARISON OF MULTIPLE JVM IMPLEMENTATIONS WITH RED HAT ENTERPRISE LINUX 6

Red Hat Enterprise Linux® 6 *and* OpenJDK

Solid platform for Java applications across multiple JVM implementations

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 the selection of a stable operating system platform an important step in defining your Java application platform. Ideally, the OS you select should be capable of accommodating and running your critical Java applications regardless of the JVM implementation used.

In the Principled Technologies labs, we compared the performance of multiple JVMs running on top of Red Hat Enterprise Linux 6, in both single-JVM and multi-JVM configurations using the SPECjbb2013 benchmark.[1]

In our tests, we found that Red Hat Enterprise Linux 6 provided a solid platform for running JVMs from Oracle and IBM, while delivering best performance on the SPECjbb2013 benchmark with OpenJDK,[2] a JVM distributed with Red Hat Enterprise Linux 6.

[1] SPECjbb* is a trademark of the Standard Performance Evaluation Corp. (SPEC). See www.spec.org for more information.
[2] OpenJDK is a trademark of Oracle, Inc.

APRIL 2013
A PRINCIPLED TECHNOLOGIES TEST REPORT
Commissioned by Red Hat, Inc.

Our results show that the Red Hat Enterprise Linux 6 operating system can provide an ideal Java application platform for intense workloads regardless of the JVM implementation choice.

COMPARING JAVA PERFORMANCE – SINGLE JVM

We used the SPECjbb2013-Composite benchmark to compare the performance of a server running a single Java Virtual machine on top Red Hat Enterprise Linux 6 using several JVM implementations. The testing included the following popular JVMs: OpenJDK, Oracle Java HotSpot, and IBM J9.[3]

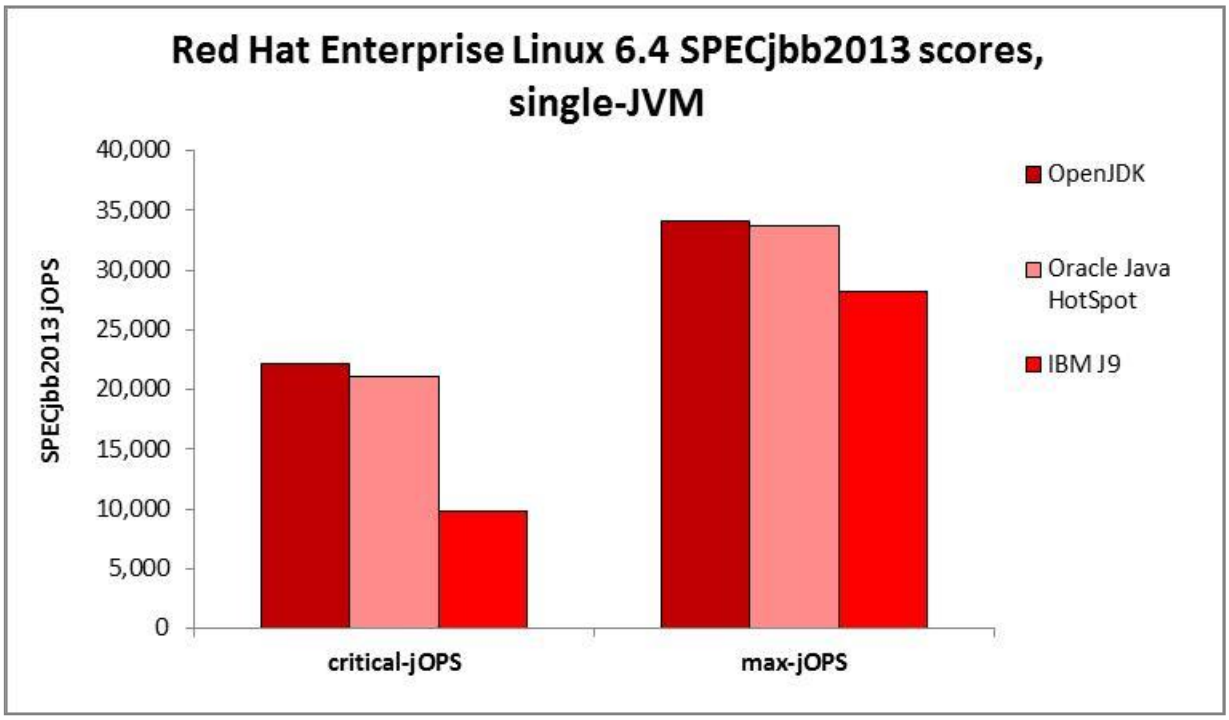

Figure 1: Composite SPECjbb2013 results comparison.

Red Hat Enterprise Linux 6		
	critical-jOPS	**max-jOPS**
OpenJDK	22,126	34,129
Oracle Java HotSpot	21,079	33,718
IBM J9	9,755	28,244
Advantage of OpenJDK over Oracle Java HotSpot	5.0%	1.2%
Advantage of OpenJDK over IBM J9	126.8%	20.8%

Figure 2: Composite SPECjbb2013 results for Red Hat Enterprise Linux 6 with various JVMs.

[3] OpenJDK test results are official results that SPEC has approved and published at www.spec.org. We did not submit Oracle and IBM JVM results to SPEC for publication.

COMPARING JAVA PERFORMANCE – MULTIPLE JVMS

We also compared the performance of a server running multiple instances of the same Java Virtual machine on top Red Hat Enterprise Linux 6 using several JVM implementations. For this series of tests, we used SPECjbb2013-MultiJVM benchmark to stress the same three JVMs: OpenJDK, Oracle Java HotSpot, and IBM J9.

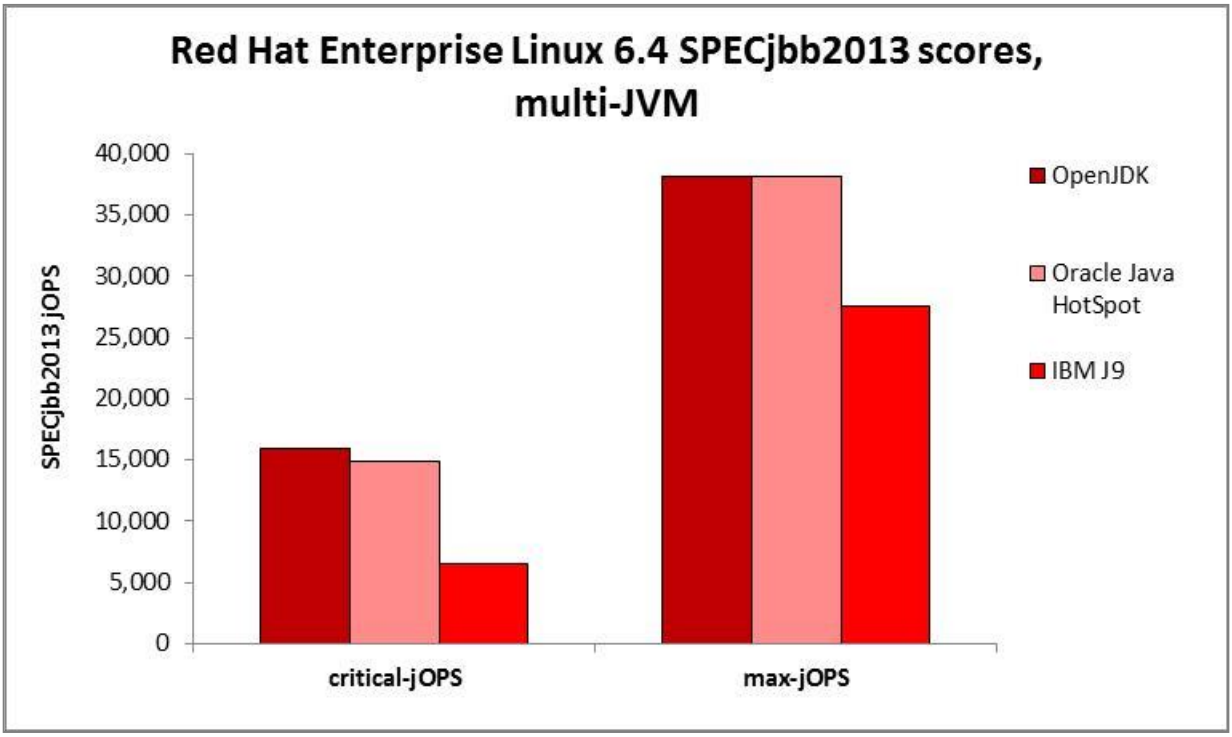

Figure 3: Multi-JVM SPECjbb2013 results comparison.

Red Hat Enterprise Linux 6		
	critical-jOPS	**max-jOPS**
OpenJDK	15,907	38,140
Oracle Java HotSpot	14,902	38,140
IBM J9	6,571	27,567
Advantage of OpenJDK over Oracle Java HotSpot	6.7%	0.0%
Advantage of OpenJDK over IBM J9	142.1%	38.4%

Figure 4: Multi-JVM SPECjbb2013 results for Red Hat Enterprise Linux 6 with various JVMs.

WHAT WE TESTED

About Red Hat Enterprise Linux 6

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. As part of its Linux offering, Red Hat includes tested open source applications. For more information about Red Hat Enterprise Linux 6, see 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 www.spec.org/jbb2013/.

IN CONCLUSION

Both the operating system and JVM implementation that you choose can 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 supported and provided a solid Java application platform for every JVM that we tested. Moreover, Red Hat Enterprise Linux 6 achieved the best performance, in both single- and multi-JVM tests, with OpenJDK, a standard JVM implementation distributed alongside the OS.

In single-JVM testing, Red Hat Enterprise Linux 6 and OpenJDK delivered a SPECjbb2013-Composite score of 22,126 critical-jOPS, the best reported critical operations score as of June 30, 2013, and outscored other popular JVMs on the SPECjbb2013-MultiJVM benchmark by as much as 38.4 percent.[4]

This study confirms that pairing Red Hat Enterprise Linux 6 with OpenJDK could provide a powerful Java environment that can help you maximize application performance.

[4] Critical-JOPS comparison to IBM J9 when running multiple VMs. See Figure 4.

APPENDIX A – SYSTEM CONFIGURATION INFORMATION

Figure 5 provides detailed configuration information for the test system.

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
Rank	Dual

System	Dell PowerEdge R720
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 5: Configuration information for our test system.

APPENDIX B - HOW WE TESTED

Installing Red Hat Enterprise Linux 6.4 and downloading SPECjbb2013

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 the JVMs

Installing OpenJDK

1. Install the OpenJDK packages using yum:
   ```
   yum –y install openjdk-1.7.0
   ```

Installing Oracle Java HotSpot

1. Download the Oracle Java SE Runtime Environment for Linux x64 from the Oracle Web site in RPM format.[5] We used the jre-7u17-linux-x64.rpm version.
2. Install using yum:
   ```
   yum –y install jre-7u17-linux-x64.rpm
   ```

Installing IBM J9

1. Download the IBM J9 package for Linux x64 from the IBM Web site in bin format. We used the ibm-java-x86_64-sdk-7.0-1.0.bin version.
2. Modify the file permissions
   ```
   chmod +x ibm-java-x86_64-sdk-7.0-4.0.bin
   ```
3. Execute the file to perform the installation.
   ```
   ./ibm-java-x86_64-sdk-7.0-1.0.bin
   ```

[5] 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 apply 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.

```
#!/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"

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

# BEST FOR IBM
```

```
#JAVA_OPTS="-Xmx240g -Xms240g -Xmn220g -Xlp -Xaggressive -XlockReservation -
Xgcpolicy:gencon -Xconcurrentlevel0 -Xgcthreads32"
#echo always > /sys/kernel/mm/redhat_transparent_hugepage/enabled


###############################################################################
# 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

# ORACLE
#JAVA=/usr/java/jre1.7.0_17/bin/java

# IBM
#JAVA=/opt/ibm/java-x86_64-70-4/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
```

ABOUT 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.