COMPARING FILE SYSTEM I/O PERFORMANCE: RED HAT ENTERPRISE LINUX 6 VS. MICROSOFT WINDOWS SERVER 2012

Red Hat® Enterprise Linux® 6

superior performance using ext4 and XFS filesystems

versus NTFS and ReFS filesystems on Microsoft® Windows Server® 2012

redhat

When choosing an operating system platform for your servers, you should know what I/O performance to expect from the operating system and file systems you select. In the Principled Technologies labs, using the IOzone file system benchmark, we compared the I/O performance of two operating systems and file system pairs, Red Hat Enterprise Linux 6 with ext4 and XFS file systems, and Microsoft Windows Server 2012 with NTFS and ReFS file systems. Our testing compared out-of-the-box configurations for each operating system, as well as tuned configurations optimized for better performance, to demonstrate how a few simple adjustments can elevate I/O performance of a file system.

We found that file systems available with Red Hat Enterprise Linux 6 delivered better I/O performance than those shipped with Windows Server 2012, in both out-of-the-box and optimized configurations. With I/O performance playing such a critical role in most business applications, selecting the right file system and operating system combination is critical to help you achieve your hardware's maximum potential.

APRIL 2013

A PRINCIPLED TECHNOLOGIES TEST REPORT

Commissioned by Red Hat, Inc.

About file system and platform configurations

While you can use IOzone to gauge disk performance, we concentrated on the file system performance of two operating systems (OSs): Red Hat Enterprise Linux 6, where we examined the ext4 and XFS file systems, and Microsoft Windows Server 2012 Datacenter Edition, where we examined NTFS and ReFS file systems. We deployed each OS in turn on the same hardware configuration, keeping server, processors, RAM, disks, RAID-groups, and other components constant. We tested the file system performance of each OS out-of-the-box and in optimized configurations that included OS and server tuning parameters.

The out-of-box configurations used the server's default BIOS settings for CPU, RAM, and system power profile (see below for more), with standard OS installation parameters. Specifically, we used the default BIOS configuration for the Dell™ PowerEdge™ R720xd, which includes the default system performance profile, called "Performance per Watt Optimized (DAPC)." This configuration setting enables the Intel processor's Turbo Boost, C States, and C1E settings, and sets the memory frequency to its maximum. The server manages CPU power.

We performed the optimized tests with the server's BIOS set to an OS-controlled system power profile. We adjusted the OS configuration to remove unnecessary processes, enabled the OS's automatic CPU or power controls, and adjusted file system parameters. For the optimized configurations, we chose the Dell Performance Per Watt Optimized (OS) system profile for our system's power setting. In the optimized configurations, the Turbo Boost settings, etc. are the same as the default profile, but with the OS managing CPU power. For the complete list of optimizations we used for both operating systems, see the Optimizing operating system configurations section of Appendix B.

BETTER DISK PERFORMANCE FOR RED HAT ENTERPRISE LINUX FILE SYSTEMS

For our testing of the four file systems, we used the IOzone Filesystem Benchmark. IOzone tests a system's file I/O performance by simulating file-access patterns that may be used in different enterprise applications, such as database or Web applications, and by using operating system-specific heuristics for reading and writing files, such as direct and asynchronous I/O, as well as operating system-specific optimizations at the file system level. We used the IOzone benchmark to test 13 synthetic file access patterns for a range of file sizes.

For each test, we first used out-of-box (default) settings, and then tested the exact same server and disk hardware with a set of tuning parameters. We ran the 13 tests as a set, and ran each set (of 13) three times for each file system and each data-

access method. For detailed system configuration information, see Appendix A. See Appendix B for systematic testing details. Appendix C and Appendix D present detailed data about our test results.

In addition, we also tested ways the OS can speed file system I/O by caching previously read or written data, or by using direct I/O methods, which seek to optimize transfer of data by using a more direct access to disk drives. To this end, we tested three OS-level I/O methods:

(1) *in-cache*, which uses the OS's file system cache

(2) out-of-cache, which uses a file size much larger than the amount of memory on the system to prevent effective file system caching throughout the majority of the test

(3) *direct I/O*, which uses the OS's implementation of direct I/O methods

Figure 1 summarizes our IOzone test results.

I/O method	Red Hat Enterprise Linux 6 (ext4 and XFS)		Microsoft Windows Server 2012 (NTFS and ReFS)		Performance delta of tested file systems	
	Out-of-box	Optimized	Out-of-box	Optimized	Out-of-box	Optimized
	ext4		NTFS		ext4 vs. NTFS	
In cache	3,960,760	5,922,869	2,398,335	4,439,896	65.2%	33.4%
Out of cache	559,697	744,479	521,545	650,378	7.3%	14.5%
Direct I/O	926,035	1,239,414	819,452	1,059,987	13.0%	16.9%
	XFS		ReFS		XFS vs. ReFS	
In cache	2,989,641	6,483,552	2,266,253	4,368,132	31.9%	48.4%
Out of cache	579,830	771,406	549,629	676,553	5.5%	14.0%
Direct I/O	863,509	1,216,965	808,590	1,055,955	6.8%	15.3%

Figure 1: IOzone results for the four file systems in KB/s.

As Figure 2 shows, using the in-cache method, both file systems we tested on Red Hat Enterprise Linux 6 delivered better performance than the file systems on Windows Server 2012, in both optimized and out-of-box configurations. For example, the ext4 file system on Red Hat Enterprise Linux 6 in out-of-box configuration delivered 65.2 percent better performance than the NTFS file system on Windows Server 2012, and the XFS file system on Red Hat Enterprise Linux 6 delivered 31.9 percent better performance than the ReFS file system on Windows Server 2012. In addition, the optimized ext4 file system on Red Hat Enterprise Linux 6 delivered 38.4 percent better performance than the optimized NTFS file system on Microsoft Windows Server 2012. Finally, the optimized XFS file system on Red Hat Enterprise Linux delivered 48.4 percent better performance than the ReFS file system on Windows Server 2012.

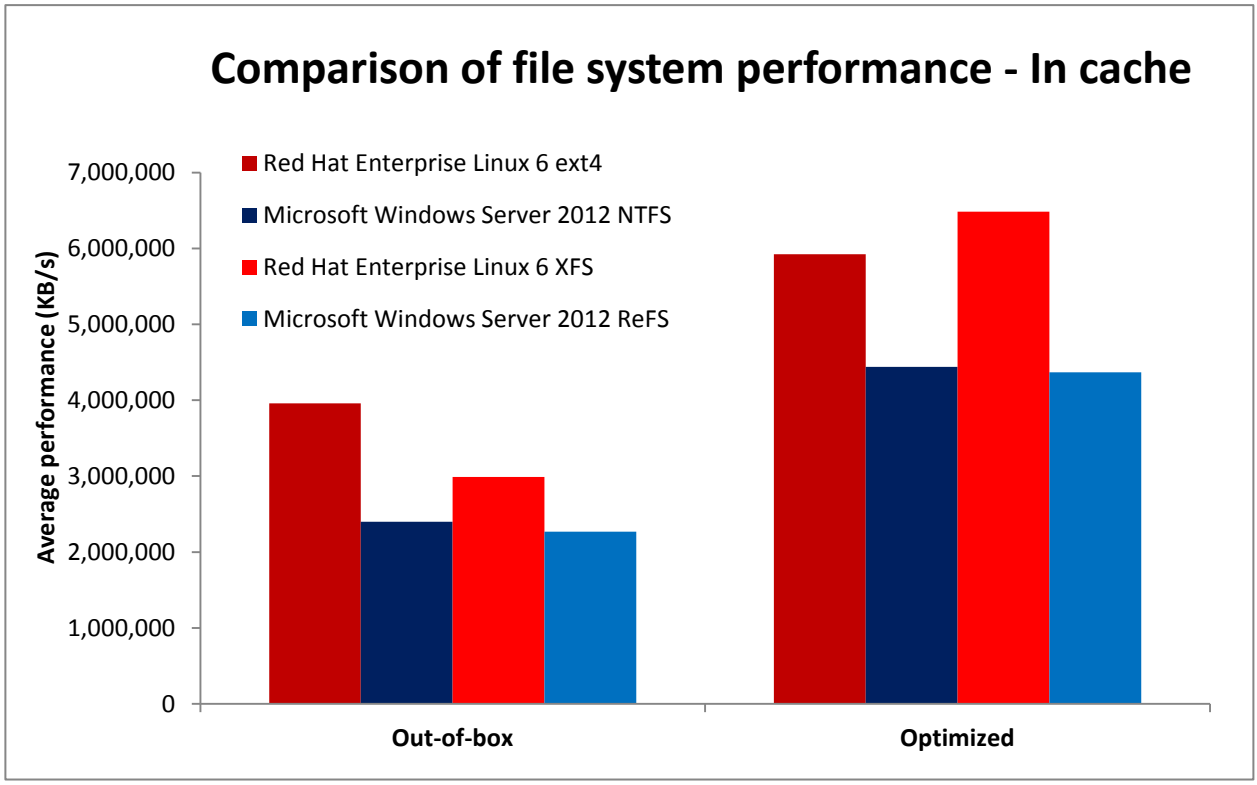

Figure 2: Comparison of the I/O performance in KB/s for the four file systems using the in-cache method. The throughput represents the geometric average of 13 IOzone tests. Higher throughput is better.

As Figure 3 shows, using the out-of-cache method, both file systems we tested on Red Hat Enterprise Linux 6 delivered better performance than the file systems on Windows Server 2012 in both out-of-box and optimized configurations. The default ext4 file system on Red Hat Enterprise Linux 6 delivered 7.3 percent better performance than the default NTFS file system on Microsoft Windows Server 2012, and the default XFS file system on Red Hat Enterprise Linux 6 delivered 5.5 percent better system performance than the default ReFS file system on Windows Server 2012. In optimized configurations, the ext4 file system on Red Hat Enterprise Linux delivered 14.5 percent better performance than the NTFS file system on Microsoft Windows Server 2012, and the optimized XFS file system on Red Hat Enterprise Linux 6 delivered 14.0 percent better performance than the optimized ReFS file system on Windows Server 2012.

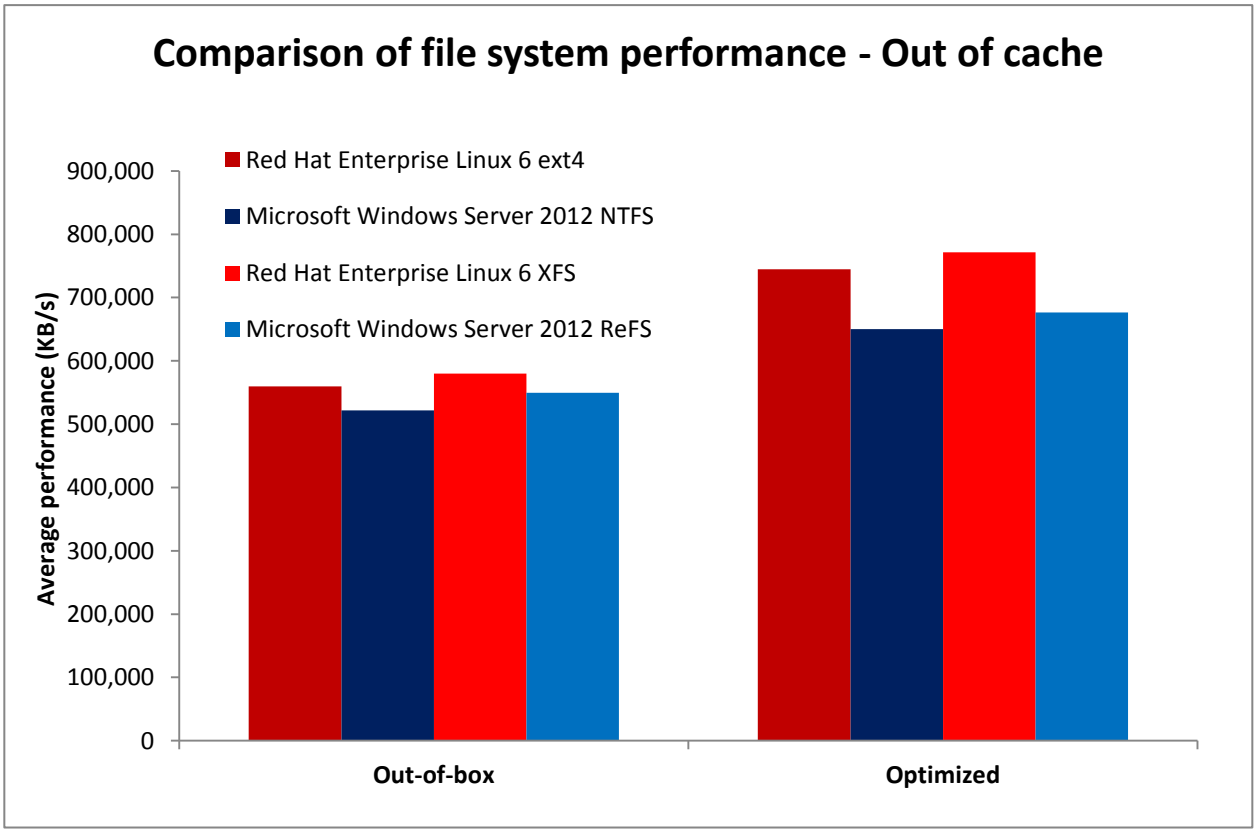

Figure 3: Comparison of the I/O performance in KB/s for the four file systems using the out-of-cache method. The throughput represents the geometric average of 13 IOzone tests. Higher throughput is better.

As Figure 4 shows, using the direct I/O method, both file systems we tested on Red Hat Enterprise Linux 6 delivered better performance than the tested file systems on Windows Server 2012 in both out-of-box and optimized configurations. The out-of-box ext4 file system on Red Hat Enterprise Linux 6 delivered 13 percent better performance than the NTFS file system on Microsoft Windows Server 2012, and the XFS file system on Red Hat Enterprise Linux 6 delivered 7.8 percent better performance than the ReFS file system on Windows Server 2012. In addition, the optimized ext4 file system on Red Hat Enterprise Linux 6 delivered 16.9 percent better performance than the optimized NTFS file system on Microsoft Windows Server 2012, and the XFS file system on Red Hat Enterprise Linux 6 delivered 15.3 percent better performance than the ReFS file system on Windows Server 2012.

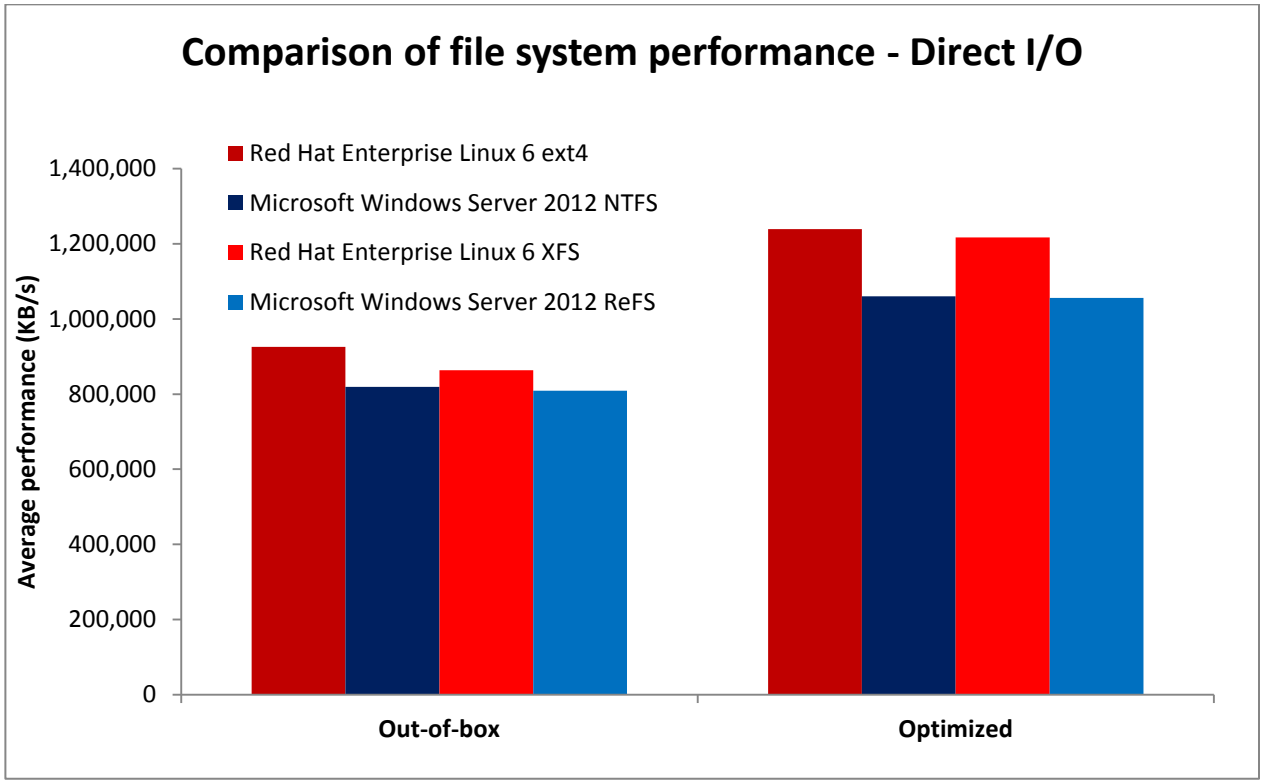

Figure 4: Comparison of the I/O performance in KB/s for the four file systems using the direct I/O method. The throughput represents the geometric average of 13 IOzone tests. Higher throughput is better.

WHAT WE TESTED

About IOzone

The IOzone benchmark tests a system's file I/O performance by simulating file-access patterns that may be used in different enterprise applications, and by using operating-system specific heuristics for reading and writing files, such as direct and asynchronous I/O, as well as operating-system specific optimizations at the file system level. The read and write operations IOzone tests include:

- Write data to a new file
- Overwrite an existing file
- Write data to random locations of a file
- Write and immediately rewrite data to a fixed section of the file
- Write data to a new file using buffered I/O system routines
- Overwrite an existing file using buffered I/O system routines
- Read an entire file
- Read an entire, recently read file
- Read the entire file starting from the file's end and proceeding to the beginning
- Read data from sections separated by a fixed amount (stride)
- Read data from random locations of a file
- Read an entire file using buffered I/O system routines
- Read an entire, recently read file using buffered I/O

For more information about IOzone, visit http://www.iozone.org.

We performed these 13 tests on files of varying sizes ranging from 1 MB to 2 GB. We also varied the record length (or size of the applications read-write buffer) from 8 KB to 1 MB in order to mimic real-world application workloads of varying sizes and kinds and to better gauge the OS's file system performance under more realistic circumstances.

About Red Hat Enterprise Linux 6

Designed to deliver performance and scalability for both small and large servers, and with documented scalability up to 4,096 CPUs and 64 terabytes of RAM, Red Hat Enterprise Linux 6 is Red Hat's flagship server operating system. 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, the operating system minimizes downtime, increases availability, and protects data due to reliability, serviceability (RAS), and scalability. Red Hat includes open source applications as part of its Linux offering. For more information about Red Hat Enterprise Linux 6, see http://www.redhat.com/f/pdf/rhel/RHEL6_datasheet.pdf.

IN CONCLUSION

Understanding how your choice of operating system affects file system I/O performance can be extremely valuable as you plan your infrastructure. Using the IOzone Filesystem Benchmark in our tests, we found I/O performance of file systems on Red Hat Enterprise Linux 6 was better than the file systems available on Microsoft Windows Server 2012, with both out-of-the-box and optimized configurations. Using default native file systems, ext4 and NTFS, we found that Red Hat Enterprise Linux 6 outperformed Windows Server 2012 by as much as 65.2 percent out-of-the-box, and as much as 33.4 percent using optimized configurations. Using more advanced native file systems, XFS and ReFS, we found that Red Hat Enterprise Linux 6 outperformed Windows Server 2012 by as much as 31.9 percent out-of-the-box, and as much as 48.4 percent using optimized configurations.

Many applications are ultimately constrained by the I/O subsystems on which they reside, making it crucial to choose the best combination of file system and operating system to achieve peak I/O performance. As our testing demonstrates, with the file system performance that Red Hat Enterprise Linux 6 can deliver, you are less likely to see I/O bottlenecks and can potentially accelerate I/O performance in your datacenter.

APPENDIX A – SYSTEM CONFIGURATION INFORMATION

Figure 5 provides detailed configuration information for the test system.

System	Dell PowerEdge R720xd
Power supplies	
Total number	2
Vendor and model number	Dell E1100E-S0
Wattage of each (W)	1100
Cooling fans	
Total number	6
Vendor and model number	AVC DBTC0638B2V
Dimensions (h x w) of each	2.5” x 2.5”
Volts	12
Amps	1.2
General	
Number of processor packages	2
Number of cores per processor	8
Number of hardware threads per core	2
System power management policy	Performance Per Watt (DAPC) or Performance Per Watt (OS) (see text)
CPU	
Vendor	Intel
Name	Xeon
Model number	E5-2660
Stepping	6
Socket type	2011LGA
Core frequency (GHz)	2.20
Bus frequency	100
L1 cache	32 KB I + 32 KB D (per core)
L2 cache	256 KB on chip (per core)
L3 cache	20 MB
Platform	
Vendor and model number	Dell PowerEdge R720xd
Motherboard model number	OM1GCR
BIOS name and version	Dell 1.5.1
BIOS settings	Default
Memory module(s)	
Total RAM in system (GB)	16
Vendor and model number	Qimonda IMSH2GE13A1F1CT13H
Type	PC3-10600
Speed (MHz)	1,333
Speed running in the system (MHz)	1,333
Timing/Latency (tCL-tRCD-tRP-tRASmin)	9-9-9-32
Size (GB)	2

System	Dell PowerEdge R720xd
Number of RAM module(s)	8
Chip organization	Double-sided
Rank	Dual
Operating system #1	
Name	Red Hat Enterprise Linux 6.4
File system	ext4 or XFS (see text)
Kernel	2.6.32-358.0.1.el6.x86_64, or 2.6.32-358.1.1.el6.x86_64 (see text)
Language	English
Operating system #2	
Name	Windows Server 2012 Datacenter Edition
Build number	9200
File system	NTFS or ReFS (see text)
Kernel	ACPI x64-based PC
Language	English
Graphics	
Vendor and model number	Matrox® G200eR
Graphics memory (MB)	16
Driver	Matrox Graphics, Inc 2.3.3.0 (8/19/2011)
RAID controller	
Vendor and model number	Dell PERC H710P Mini
Firmware version	21.1.0-007
Cache size	1 GB
RAID configuration	OS #1 boot volume: RAID 1 configuration of two disks (Hard drive type #3) OS #2 boot volume: RAID 1 configuration of two disks (Hard drive type #2) IOzone test volume: RAID 0 configuration of 17 disks (Hard drive type #1) OS swap volume: RAID 0 configuration of three disks (Hard drive type #1)
Hard drives type #1	
Vendor and model number	Dell MBF2600RC
Number of drives	20
Size (GB)	600
Buffer size (MB)	16
RPM	10K
Type	SAS
Hard drives type #2	
Vendor and model number	Fujitsu MBB2073RC
Number of drives	2
Size (GB)	73

System	Dell PowerEdge R720xd
Buffer size (MB)	16
RPM	10K
Type	SAS
Hard drives type #3	
Vendor and model number	Dell Savvio ST9146803SS
Number of drives	2
Size (GB)	146
Buffer size (MB)	16
RPM	10K
Type	SAS
Ethernet adapters	
Vendor and model number	Intel Gigabit 4P I350-t rNDC
Type	Internal
Optical drive(s)	
Vendor and model number	TEAC DV-28SW
Type	DVD-ROM
USB ports	
Number	4 external, 1 internal
Type	2.0

Figure 5: Configuration information for our test system.

APPENDIX B - HOW WE TESTED

Red Hat Enterprise Linux 6.4: Installation, configurations, and IOzone testing

Installing Red Hat Enterprise Linux 6.4

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. Install the XFS package:

```
yum install xfsprogs
```

24. Create partitions on the IOzone and swap disks (here /dev/sdb and /dev/sdd, respectively):

```
parted /dev/sdb mklabel gpt
parted /dev/sdb mkpart primary "1 -1"
parted /dev/sdb name 1 Iozone
parted /dev/sdd mklabel gpt
parted /dev/sdd mkpart primary linux-swap "1 -1"
parted /dev/sdd name 1 Swap
```

25. Create an ext4 or XFS file system on the IOzone partition (here /dev/sdb1) depending on the test:

```
# Either an ext4 filesystem
mkfs.ext4 /dev/sdb1
# or XFS file system
mkfs.xfs -f /dev/sdb1
```

26. Mount the Iozone test disk at /test:

```
mkdir /test
mount /dev/sdb1 /test
```

27. Create a swap area on the new swap partition (here /dev/sdd1):

```
mkswap /dev/sdd1
swapoff -a
swapon /dev/sdd1
```

Installing the IOzone software on Red Hat Enterprise Linux 6.4

1. Log onto Red Hat Enterprise Linux 6.4.
2. Install the GCC compiler as well as the run-time libraries for 32-bit programs by adding the following packages: gcc, libc.i686, libgcc.i686, libstdc++.i686, and glibc-devel.i686.
3. Download the IOzone 3.414 source code from www.iozone.org.

```
wget http://www.iozone.org/src/current/iozone3_414.tar
```

4. Un-tar the source code and go to the main directory:

```
tar xf iozone3_414.tar
cd iozone3_414/src/current/
```

5. Modify the makefile to force 32-bit compilation by applying this patchfile:

```
patch < makefile-patch

The patch file for IOzone's makefile (makefile-patch) is
      diff -u makefile*
      --- makefile    2013-03-20 16:51:15.559646000 -0400
      +++ makefile-   2013-03-20 16:46:51.007276100 -0400
      @@ -9,7 +9,7 @@
       #              convex, FreeBSD, OpenBSD, OSFV3, OSFV4, OSFV5, SCO
       #              SCO_Unixware_gcc,NetBSD,TRU64, Mac OS X

      -CC     = cc -m32
      +CC     = cc
       C89    = c89
       GCC    = gcc
       CCS    = /usr/ccs/bin/cc
      @@ -1220,10 +1220,10 @@
              @echo ""
              @echo "Building iozone for Windows (No async I/O)"
              @echo ""
      -       $(GCC) -c -O3 -Dunix -DHAVE_ANSIC_C -DNO_MADVISE  \
      -              -DWindows $(CFLAGS) iozone.c \
      +       $(GCC) -c -O -Dunix -DHAVE_ANSIC_C -DNO_MADVISE  \
      +              -DWindows $(CFLAGS) -DDONT_HAVE_O_DIRECT iozone.c \
                     -o iozone_windows.o
              $(GCC) -c -O -Dunix -DHAVE_ANSIC_C -DNO_MADVISE \
                     -DWindows $(CFLAGS) libbif.c -o libbif.o
```

6. Create the iozone binary, `iozone`.

```
make linux
```

Running the IOzone tests on Red Hat Enterprise Linux 6.4

The following three bash scripts are used to perform IOzone tests for the corresponding file-access methods: direct I/O, in-cache, and out-of-cache.

dio.sh

```
#!/bin/bash
## Direct-I/O Method, March 2013
## First argument is added the output file
mkdir /test/$1 > /dev/null 2>&1
./iozone -n 1024 -g 4096000 -y 8 -q 1024 -a -R -I \
   -f /test/$1/t1 > directio_$1.txt
rm -f /test/$1/t1
echo 3 > /proc/sys/vm/drop_caches
sync
## End of the Direct-I/O script
```

inc.sh

```
#!/bin/bash
## In-Filesystem-Cache Method, March 2013
## First argument is added the output file
mkdir /test/$1 > /dev/nul 2>&1
./iozone -n 1024 -g 4096000 -y 8 -q 1024 -a -R \
   -f /test/$1/t1 > incache_$1.txt
rm -f /test/$1/t1
echo 3 > /proc/sys/vm/drop_caches
sync
## End of the In-Filesystem-Cache script
```

otc.sh

```
#!/bin/bash
## Out-of-Filesystem-Cache Method, March 2013
## First argument is added the output file
mkdir /test/$1 >/dev/null 2>&1
for r in 64k 1024k; do
  ./iozone -s 32g -r $r -C -c -e -w -x \
      -f /test/$1/t1 > outcache_$1-$r.txt
  rm -f /test/$1/t1
  echo 3 > /proc/sys/vm/drop_caches
  sync
done
## End of the Out-of-Filesystem-Cache script
```

Microsoft Windows Server 2012 Datacenter: Installation, configurations, and IOzone testing

Installing Microsoft Windows Server 2012 Datacenter

1. Insert and boot from the Windows Server 2012 Datacenter installation DVD.
2. At the first Window Setup screen, keep the defaults for installation language, time/currency format, and keyboard input method. Click Next.
3. At the second Windows Setup screen, click Install now.
4. At the third Windows Setup screen, enter the Windows activation key, and click Next.
5. At the fourth Windows Setup screen, select the Windows Server 2012 Datacenter (Server with a GUI), and click Next.
6. At the fifth Windows Setup screen, select the checkbox to accept the license term, and click Next.
7. At the sixth Windows Setup screen, click Custom: Install Windows only (advanced).
8. At the seventh Windows Setup screen, select Drive 2 as the Windows installation drive, and click Next to start installation.

9. The system will reboot. At the Settings screen, enter the password for the Administrator (twice), and click Finish.
10. Log in as administrator.
11. Open the Server Manager.
12. Select File and Storage Services.
13. Select Disks.
14. From the list of disks, right-click on the IOzone device, select Reset Disk, and click Yes to erase the data.
15. From the list of disks, right-click on the IOzone device, and select New Volume…
16. On the Server and Disk screen, select the disk, and click Next.
17. Click OK to initialize the disk with a GPT label.
18. On the Size screen, keep the defaults, and click Next.
19. On the Drive Letter or Folder screen, Select The following folder, and enter c:\test. Click OK to create this folder.
20. On the File System Settings screen, select NTFS or ReFS, depending on the file system under test.
21. On the same screen, enter a Volume label of IOzone, and click Next.
22. On the Confirmation screen, click Create.
23. Create a volume for swap following steps 12-22 with the location changed to an unused drive letter, and the file system type as NTFS for both tests.
24. Close the Server Manager.
25. From Explorer, right-click Computer, and select Properties.
26. From the System Control Panel, click Advanced Settings.
27. Under Performance, click Settings.
28. Select the Advanced tab.
29. Under Virtual Memory, click Change.
30. On the Virtual Memory screen, select D: (the new swap drive), click Custom Size, and enter the free space on size the drive less 10 MB for both Initial size (MB) and Maximum size (MB). Click Set.
31. On the Virtual Memory screen, select C: (the boot drive), click No paging file, and click Set.
32. On the Virtual Memory screen, click Ok.
33. Close all screens, clicking OK as needed, and restart the server.

Installing the IOzone software on Windows Server 2012

The IOzone software uses Unix/Linux style APIs for file system access. Creating a version for a Windows system uses the Cygwin environment.

1. Download the IOzone 3.414 source code from www.iozone.org.

```
wget http://www.iozone.org/src/current/iozone3_414.tar
```

2. Un-tar the source code and go to the main directory:

```
tar xf iozone3_414.tar
cd iozone3_414/src/current/
```

3. Modify the makefile to force 32-bit compilation by applying this patchfile:

```
patch < makefile-patch

The patch file for IOzone's makefile (makefile-patch)is
      diff -u makefile*
```

```
--- makefile    2013-03-20 16:51:15.559646000 -0400
+++ makefile-   2013-03-20 16:46:51.007276100 -0400
@@ -9,7 +9,7 @@
 #               convex, FreeBSD, OpenBSD, OSFV3, OSFV4, OSFV5, SCO
 #               SCO_Unixware_gcc,NetBSD,TRU64, Mac OS X

-CC      = cc -m32
+CC      = cc
 C89     = c89
 GCC     = gcc
 CCS     = /usr/ccs/bin/cc
@@ -1220,10 +1220,10 @@
         @echo ""
         @echo "Building iozone for Windows (No async I/O)"
         @echo ""
-        $(GCC) -c -O3 -Dunix -DHAVE_ANSIC_C -DNO_MADVISE  \
-                -DWindows $(CFLAGS) iozone.c \
+        $(GCC) -c -O -Dunix -DHAVE_ANSIC_C -DNO_MADVISE  \
+                -DWindows $(CFLAGS) -DDONT_HAVE_O_DIRECT iozone.c \
                 -o iozone_windows.o
         $(GCC) -c -O -Dunix -DHAVE_ANSIC_C -DNO_MADVISE \
                 -DWindows $(CFLAGS) libbif.c -o libbif.o
```

4. Create the iozone binary, `iozone.exe`.

   ```
   make Windows
   ```

5. Copy the IOzone binary and the Cygwin DLL, /bin/cygwin1.dll from the build server to the Windows server under test.

Running the IOzone tests on Windows Server 2012

The following three batch scripts are used to perform IOzone tests for the corresponding file-access methods: direct I/O, in-cache, and out-of-cache.

dio.bat

```
rem ## IOzone with Direct I/O, March 2013
rem ## the first argument is added to the run's output file
del \test\t1 > NUL 2>&1
.\iozone.exe -n 1024 -g 4096000 -y 8 -q 1024 -a -R -I -f \test\t1 > dio-%1%.txt
del \test\t1
shutdown /r
rem ## end of Direct-I/O script
```

inc.bat

```
rem ## IOzone with the In-Filesystem-Cache method, March 2013
rem the first argument is added to the run's output file
del \test\t1 > NUL 2>&1
.\iozone.exe -n 1024 -g 4096000 -y 8 -q 1024 -a -R -f \test\t1 > inc-%1%.txt
del \test\t1
shutdown /r
rem ## end of In-Filesystem-Cache script
```

out.bat

```
rem ## IOzone with the Out-of-Filesystem-Cache method, March 2013
rem ## the first argument is added to the run's output file
del \test\t1 > NUL 2>&1
.\iozone.exe -s 32g -r 64k -C -c -e -w -x -f \test\t1 > out-%1%-64k.txt
```

```
del \test\t1
.\iozone.exe -s 32g -r 1024k -C -c -e -w -x -f \test\t1 > out-%1%-1024k.txt
del \test\t1
shutdown /r
rem ## end of Out-of-Filesystem-Cache script
```

Optimizing operating system configurations

Before running IOzone for the optimized Red Hat Enterprise Linux configuration, run the following two bash scripts. Red Hat Enterprise Linux 6.4 uses the tuned utility with the enterprise-storage profile to configure the file systems for better performance and to run the CPUs at high performance.

AdditionalLinuxFileSystemTuning.sh

```
#!/bin/bash
## For the optimized-configuration tests, ensure the filesystem
## under test is mounted without journal write-barriers
## March 2013
mount /test
swapon /dev/sdd1
tuned-adm profile default
tuned-adm profile enterprise-storage
mount -o remount,barrier=0 /test
cat /proc/mounts
swapon -s
## End of AdditionalLinuxFileSystemTuning.sh
```

DisableSomeDefaultServices.sh

```
#!/bin/bash
## For the optimized-configuration tests, disable unneeded services
## March 2013
for i in abrt-ccpp abrt-oops abrtd acpid atd auditd autofs \
    avahi-daemon cgconfig crond cups haldaemon irqbalance kdump\
    libvirt-guests mcelogd mdmonitor messagebus portreserve\
    postfix rhnsd rhsmcertd rpcbind rpcgssd rpcidmapd certmonger\
    netfs sysstat; do
  service $i stop
done
service lvm2-monitor force-stop
## end of DisableSomeDefaultServices.sh
```

Before running IOzone for the optimized Windows Server 2012 configuration, run the following batch script. In particular, the OS power profile is set to High performance and the desktop GUI is configured for high performance.

CommandsNoPersonaManagement.bat

```
rem Note: script closely adapted from
rem http://mtellin.com/2010/09/13/creating-a-windows-7-template-for-vmware-view/
rem Version dated 2012-02-05
reg load "hku\temp" "%USERPROFILE%\..\Default User\NTUSER.DAT"
reg ADD "hku\temp\Software\Policies\Microsoft\Windows\Control Panel\Desktop" /v
SCRNSAVE.EXE /d "%windir%\system32\scrnsave.scr" /f
reg ADD "hku\temp\Software\Policies\Microsoft\Windows\Control Panel\Desktop" /v
ScreenSaveTimeOut /d "600" /f
reg ADD "hku\temp\Software\Policies\Microsoft\Windows\Control Panel\Desktop" /v
ScreenSaverIsSecure /d "1" /f
```

```
reg ADD "hku\temp\Software\Microsoft\Windows\CurrentVersion\Policies\System" /v
Wallpaper /d " " /f
reg ADD "hku\temp\Software\Microsoft\Windows\CurrentVersion\Internet
Settings\Cache" /v Persistent /t REG_DWORD /d 0x0 /f
reg ADD "hku\temp\Software\Microsoft\Feeds" /v SyncStatus /t REG_DWORD /d 0x0 /f
reg ADD "hku\temp\Software\Microsoft\WIndows\CurrentVersion\Policies\Explorer" /v
HideSCAHealth /t REG_DWORD /d 0x1 /f
reg unload "hku\temp"

reg ADD "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Internet Explorer\Main" /v
DisableFirstRunCustomize /t REG_DWORD /d 0x1 /f
reg ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory
Management\PrefetchParameters" /v EnableSuperfetch /t REG_DWORD /d 0x0 /f
reg ADD "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU"
/v NoAutoUpdate /t REG_DWORD /d 0x1 /f
reg ADD "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\SystemRestore"
/v DisableSR /t REG_DWORD /d 0x1 /f
reg ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Disk" /v TimeOutValue
/t REG_DWORD /d 200 /f
reg ADD "HKEY_LOCAL_MACHINE\SOFTWARE\Image" /v Revision /t REG_SZ /d 1.0 /f

reg ADD "HKEY_LOCAL_MACHINE\SOFTWARE\Image" /v Virtual /t REG_SZ /d Yes /f
reg ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\Application"
/v MaxSize /t REG_DWORD /d 0x100000 /f
reg ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\Application"
/v Retention /t REG_DWORD /d 0x0 /f
reg ADD
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Network\NewNetworkWindowOff"
/f
reg ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\System" /v
MaxSize /t

REG_DWORD /d 0x100000 /f
reg ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\System" /v
Retention /t REG_DWORD /d 0x0 /f
reg ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\Security" /v
MaxSize /t REG_DWORD /d 0x100000 /f
reg ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\Security" /v
Retention /t REG_DWORD /d 0x0 /f
reg ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\CrashControl" /v
CrashDumpEnabled /t REG_DWORD /d 0x0 /f
reg ADD
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer" /v
NoRecycleFiles /t REG_DWORD /d 0x1 /f
reg ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v
fDenyTSConnections /t REG_DWORD /d 0x0 /f
reg ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal
Server\WinStations\RDP-Tcp" /v UserAuthentication /t REG_DWORD /d 0x0 /f
reg ADD
"HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\policies\system" /v
EnableLUA /t REG_DWORD /d 0x0 /f
reg Add "HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\Sideshow" /v
Disabled /t REG_DWORD /d 0x1 /f

Powershell Set-Service 'BDESVC' -startuptype "disabled"
Powershell Set-Service 'wbengine' -startuptype "disabled"
Powershell Set-Service 'DPS' -startuptype "disabled"
```

```
Powershell Set-Service 'UxSms' -startuptype "disabled"
Powershell Set-Service 'Defragsvc' -startuptype "disabled"
Powershell Set-Service 'HomeGroupListener' -startuptype "disabled"
Powershell Set-Service 'HomeGroupProvider' -startuptype "disabled"
Powershell Set-Service 'iphlpsvc' -startuptype "disabled"
Powershell Set-Service 'MSiSCSI' -startuptype "disabled"
Powershell Set-Service 'swprv' -startuptype "disabled"
Powershell Set-Service 'CscService' -startuptype "disabled"
Powershell Set-Service 'SstpSvc' -startuptype "disabled"
Powershell Set-Service 'wscsvc' -startuptype "disabled"
Powershell Set-Service 'SSDPSRV' -startuptype "disabled"
Powershell Set-Service 'SysMain' -startuptype "disabled"
Powershell Set-Service 'TabletInputService' -startuptype "disabled"
Powershell Set-Service 'Themes' -startuptype "disabled"
Powershell Set-Service 'upnphost' -startuptype "disabled"
Powershell Set-Service 'VSS' -startuptype "disabled"
Powershell Set-Service 'SDRSVC' -startuptype "disabled"
Powershell Set-Service 'WinDefend' -startuptype "disabled"
Powershell Set-Service 'WerSvc' -startuptype "disabled"
Powershell Set-Service 'MpsSvc' -startuptype "disabled"
Powershell Set-Service 'ehRecvr' -startuptype "disabled"
Powershell Set-Service 'ehSched' -startuptype "disabled"
Powershell Set-Service 'WSearch' -startuptype "disabled"
Powershell Set-Service 'wuauserv' -startuptype "disabled"
Powershell Set-Service 'Wlansvc' -startuptype "disabled"
Powershell Set-Service 'WwanSvc' -startuptype "disabled"
bcdedit /set BOOTUX disabled
vssadmin delete shadows /All /Quiet
Powershell disable-computerrestore -drive c:\
netsh advfirewall set allprofiles state off
powercfg -H OFF
powercfg -setactive 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c
net stop "sysmain"
fsutil behavior set DisableLastAccess 1
schtasks /change /TN "\Microsoft\Windows\Defrag\ScheduledDefrag" /Disable
schtasks /change /TN "\Microsoft\Windows\SystemRestore\SR" /Disable
schtasks /change /TN "\Microsoft\Windows\Registry\RegIdleBackup" /Disable
schtasks /change /TN "\Microsoft\Windows Defender\MPIdleTask" /Disable
schtasks /change /TN "\Microsoft\Windows Defender\MP Scheduled Scan" /Disable
schtasks /change /TN "\Microsoft\Windows\Maintenance\WinSAT" /Disable
rem End of CommandsNoPersonaManagement.bat
```

Analyzing IOzone results

Iozone writes its data in fields of fixed width, and when the reported numbers are high, fields may run into each other. The following bash script reformats the data so that the 13 fields are separated by one space.

fix-iozone-fields.sh

```
#!/bin/bash
## Auxiliary script to reformat IOzone's main table (data transfer
## speed for each file size and ## record length pair) when transfer
## rates are so high adjacent columns abut.
## March 2013

## Run as a Unix-style filter
```

```
cut -c1-16,17-24,25-32,33-40,41-49,40-58,59-66,67-74,75-82,83-91,92-100,101-
109,110-118,119-126,127-135 --output-delimiter=" "
## end of fix-iozone-fields script
```

APPENDIX C – DETAILED RESULTS CHARTS

Figures 7 through 22 chart our IOzone test results, for each OS configuration (out-of-the-box and optimized), file system (ext4, XFS, NTFS, and ReFS) and file access method (in-cache, direct I/O, and out-of-cache). The charts present the average file system performance in KB/s over the 13 IOzone subtests, plotted for file size from 1,024 KB to 2,097,152 KB and record lengths from 8 KB to 1,024 KB. Note that there are no charts for the out-of-cache method, because by design it returns only two data points per test.

For numerical data corresponding to each chart, see Appendix D.

Figure 6 summarizes the results of the IOzone tests.

I/O Method	Red Hat Enterprise Linux 6.4 (ext4 and XFS)		Microsoft Windows Server 2012 (NTFS and ReFS)		Performance delta of tested file systems	
	Out-of-box	Optimized	Out-of-box	Optimized	Out-of-box	Optimized
	ext4		NTFS		ext4 vs. NTFS	
In cache	3,960,760	5,922,869	2,398,335	4,439,896	65.2%	33.4%
Out of cache	559,697	744,479	521,545	650,378	7.3%	14.5%
Direct I/O	926,035	1,239,414	819,452	1,059,987	13.0%	16.9%
	XFS		ReFS		XFS vs. ReFS	
In cache	2,989,641	6,483,552	2,266,253	4,368,132	31.9%	48.4%
Out of cache	579,830	771,406	549,629	676,553	5.5%	14.0%
Direct I/O	863,509	1,216,965	808,590	1,055,955	6.8%	15.3%

Figure 6: IOzone results for the four file systems in KB/s.

In cache charts

ext4

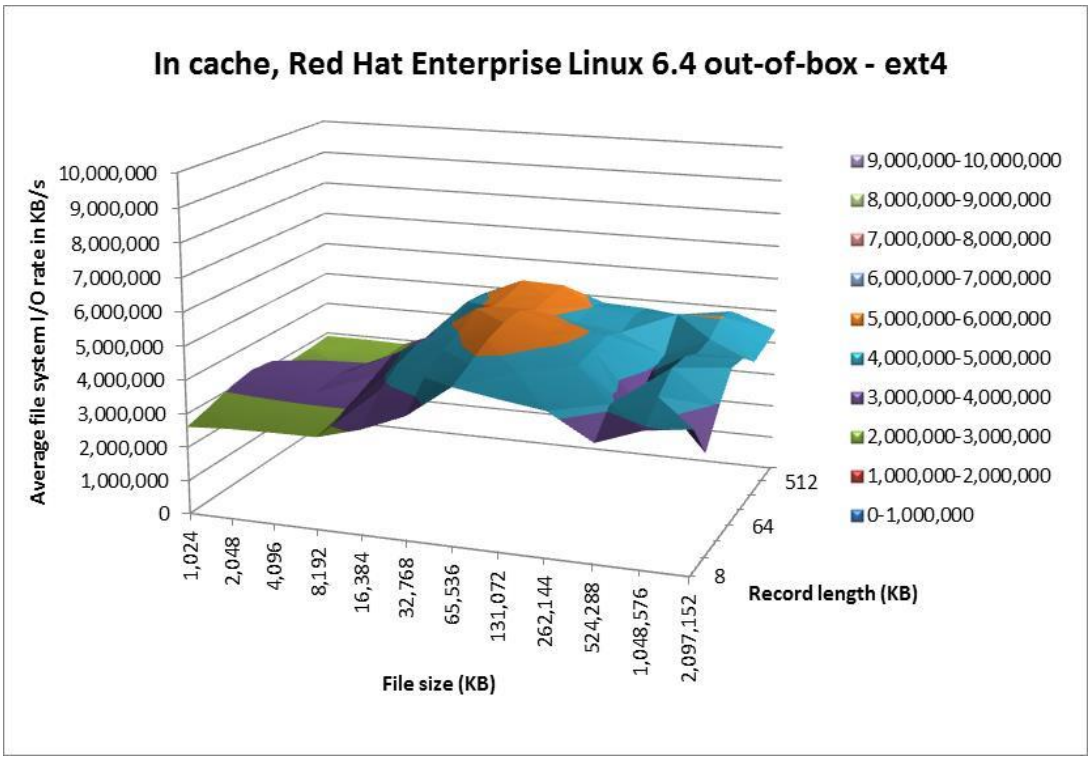

Figure 7: Average IOzone performance in KB/s for ext4 file system on out-of-box Red Hat Enterprise Linux 6.4 platform with the in-cache method.

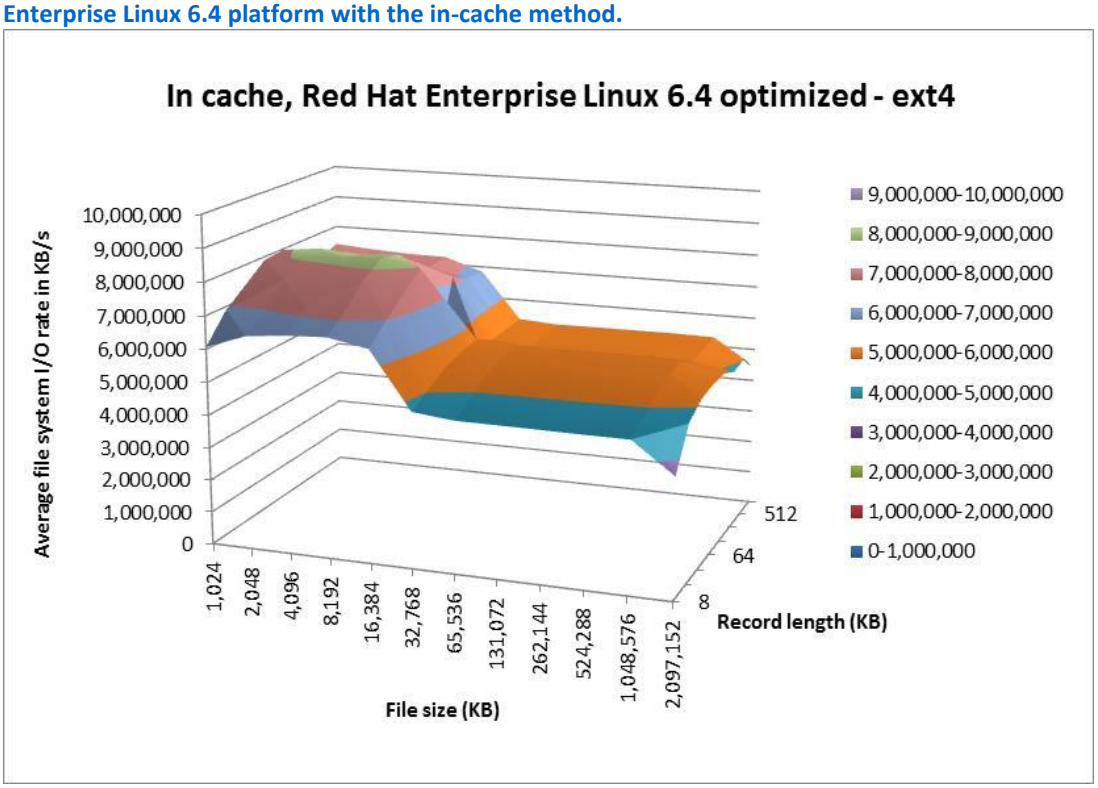

Figure 8: Average IOzone performance in KB/s for ext4 file system on optimized Red Hat Enterprise Linux 6.4 platform with the in-cache method.

NTFS

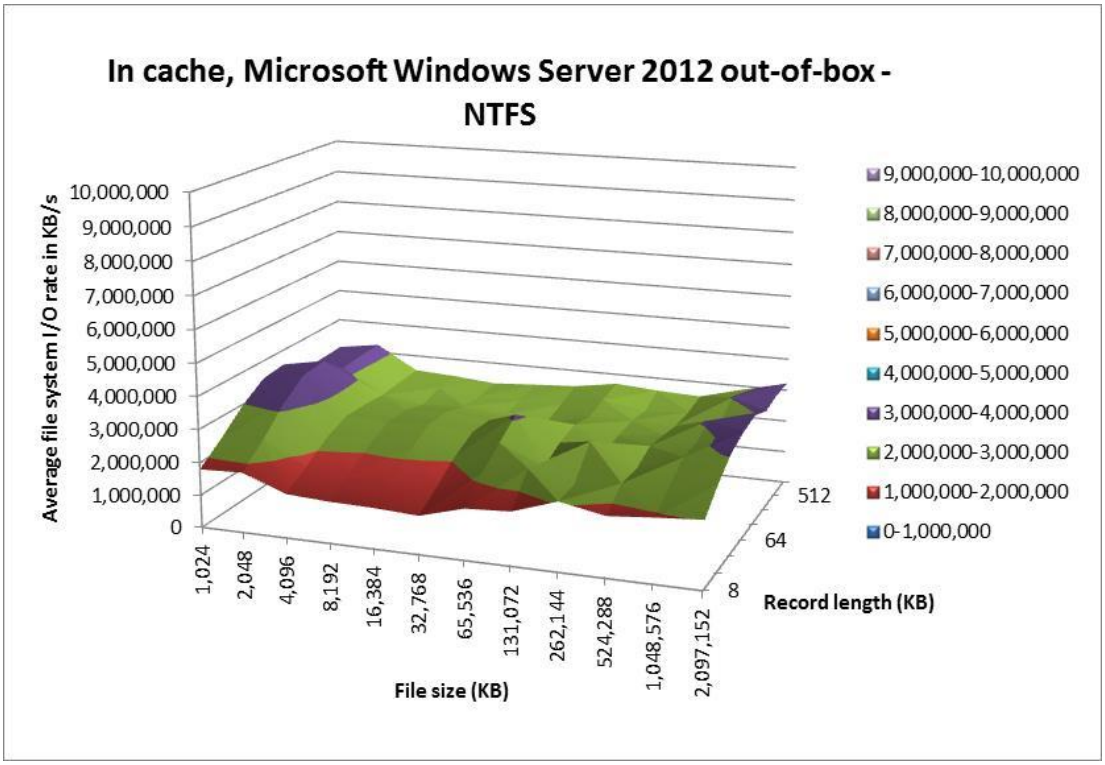

Figure 9: Average IOzone performance in KB/s for NTFS file system on out-of-box Microsoft Windows Server 2012 with the in-cache method.

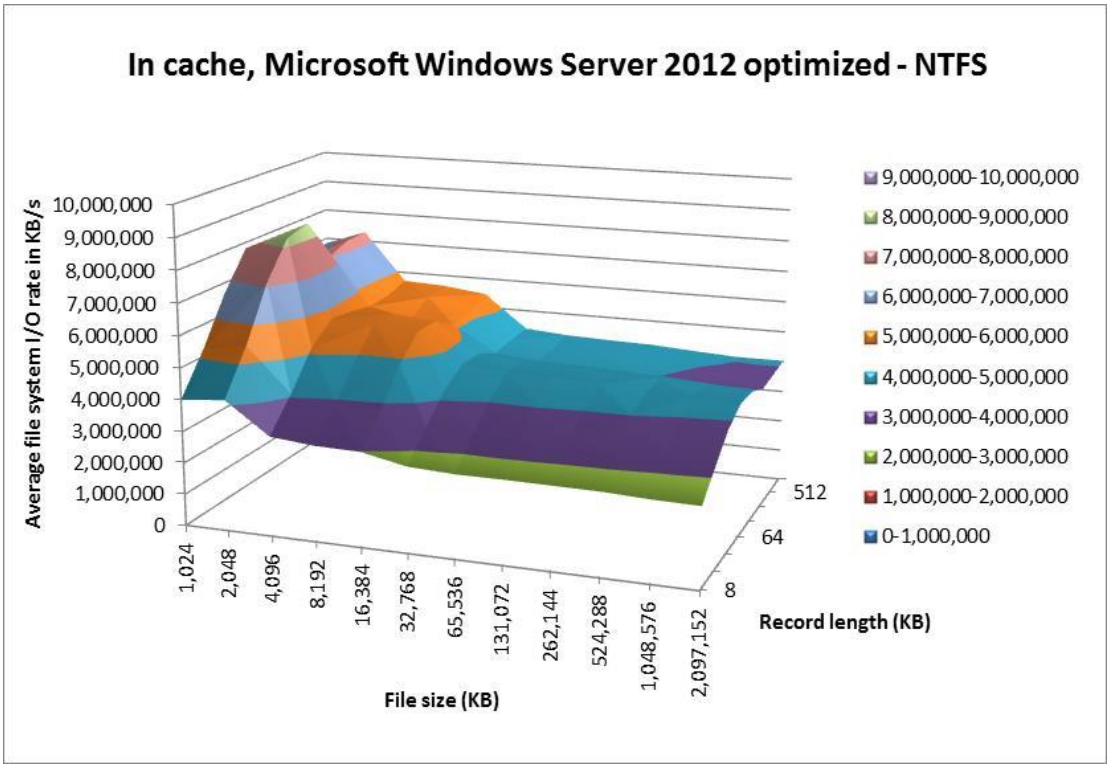

Figure 10: Average IOzone performance in KB/s for NTFS file system on optimized Microsoft Windows Server 2012 with the in-cache method.

XFS

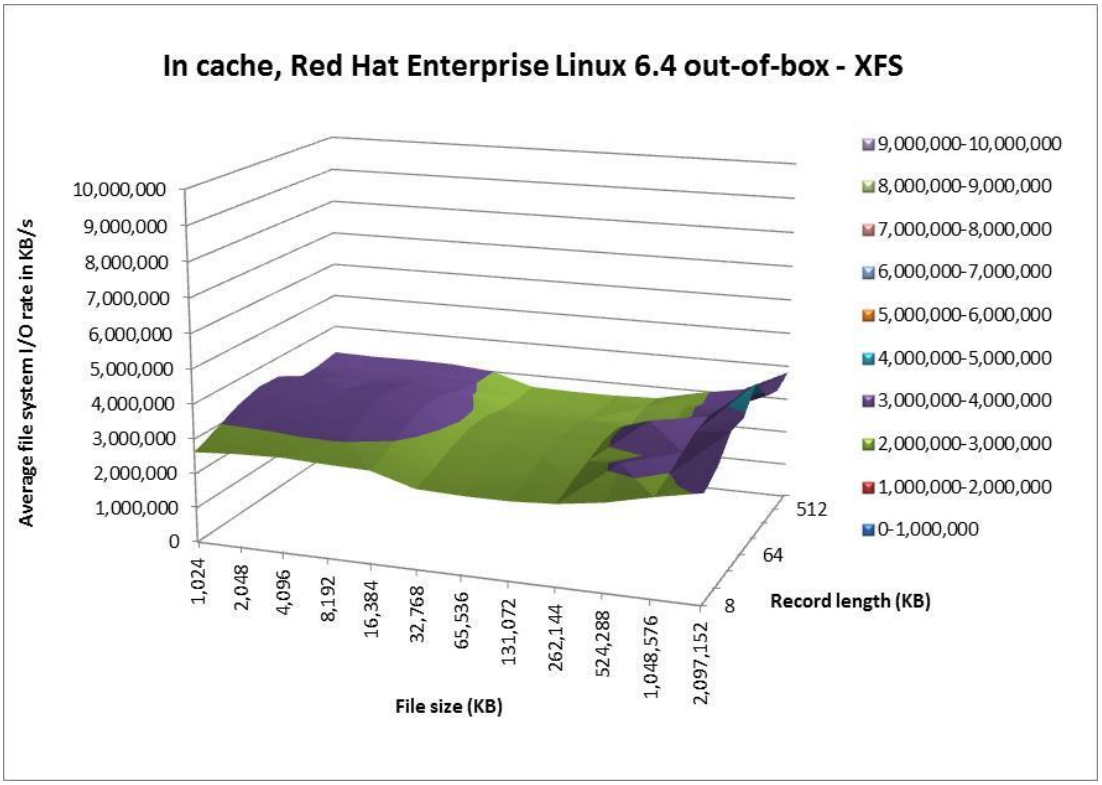

Figure 11: Average IOzone performance in KB/s for XFS file system on out-of-box Red Hat Enterprise Linux 6.4 platform with the in-cache method.

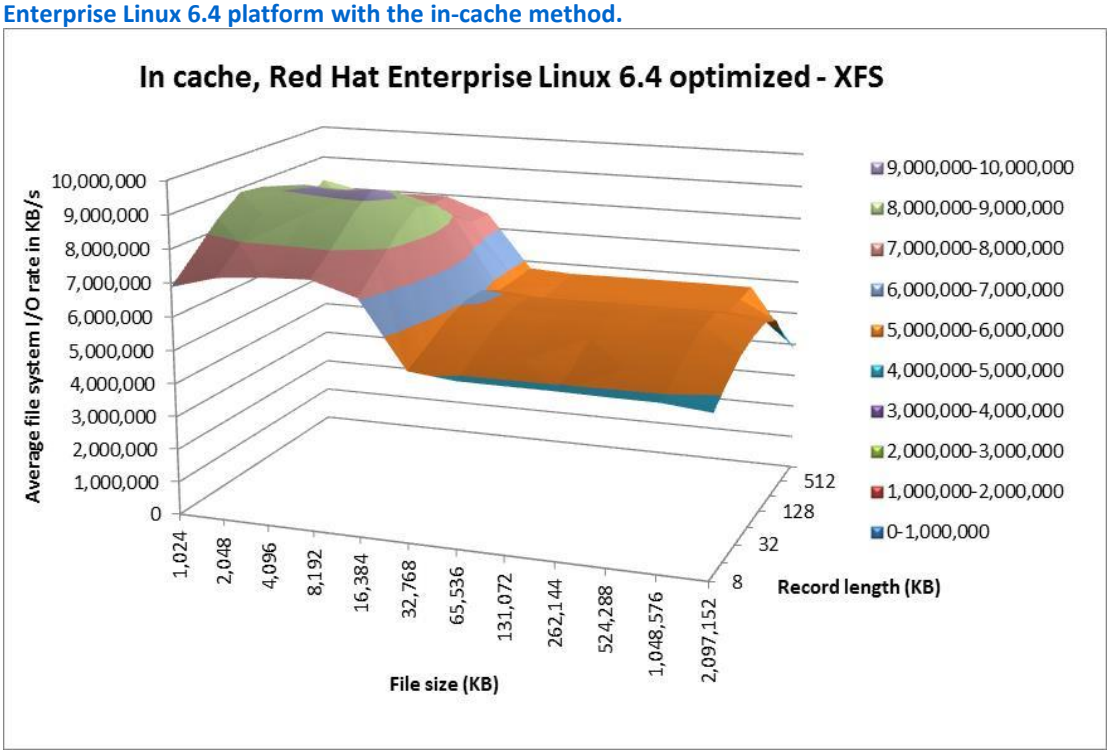

Figure 12: Average IOzone performance in KB/s for XFS file system on optimized Red Hat Enterprise Linux 6.4 platform with the in-cache method.

ReFS

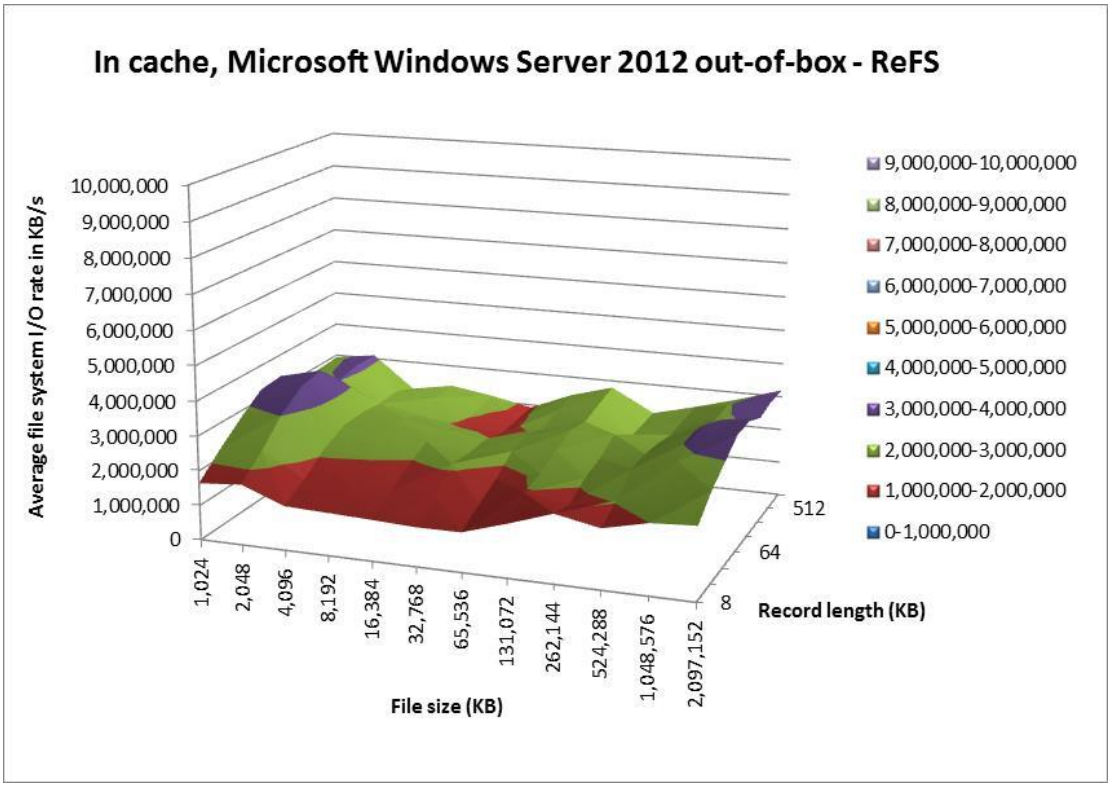

Figure 13: Average IOzone performance in KB/s for ReFS file system on out-of-box Microsoft Windows Server 2012 with the in-cache method.

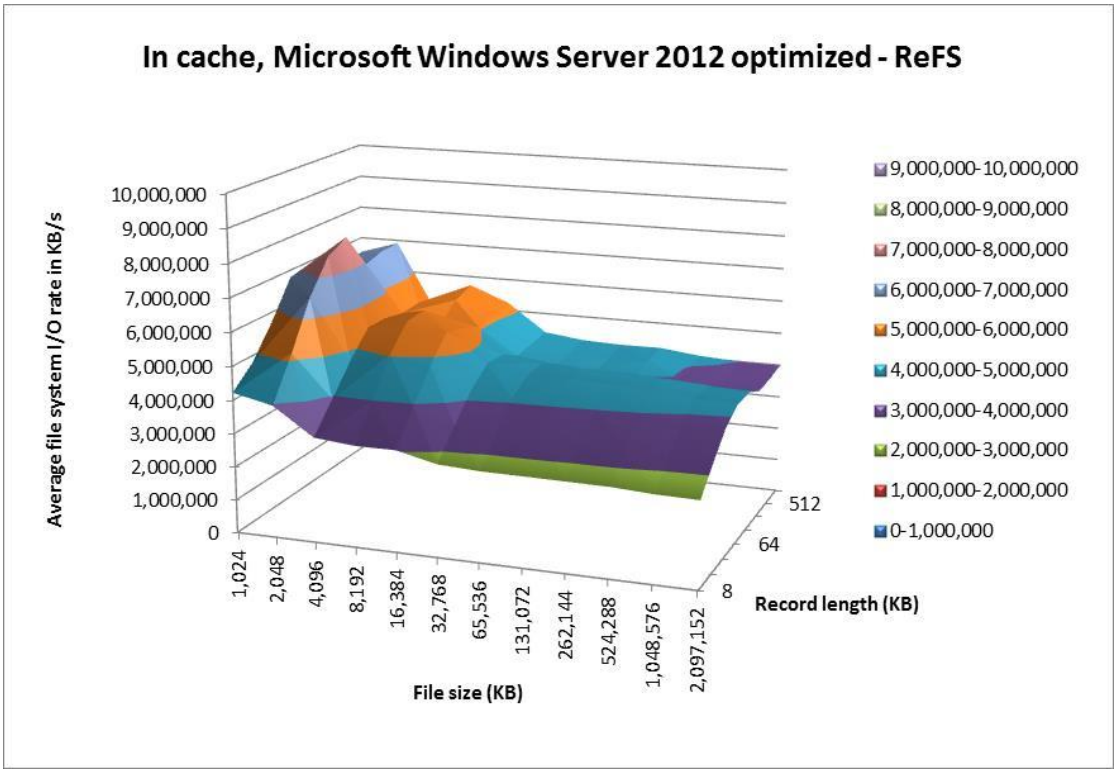

Figure 14: Average IOzone performance in KB/s for ReFS file system on optimized Microsoft Windows Server 2012 with the in-cache method.

Direct I/O charts

ext4

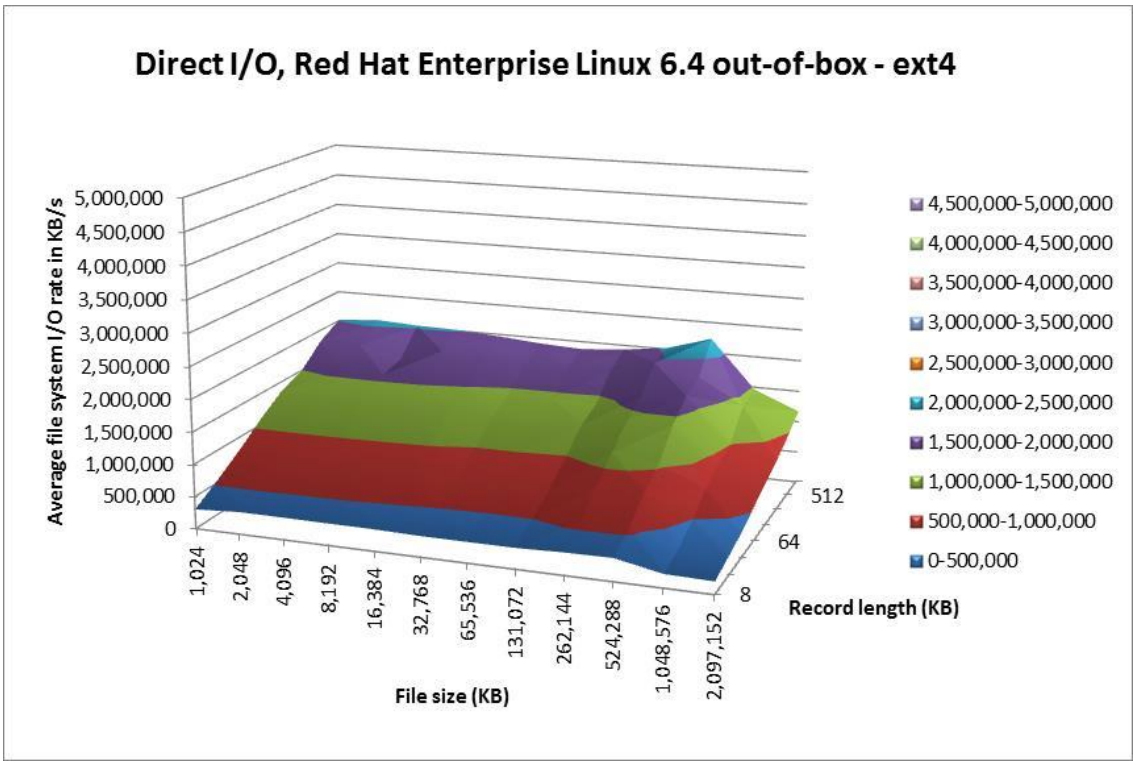

Figure 15: Average IOzone performance in KB/s for ext4 file system on out-of-box Red Hat Enterprise Linux 6.4 platform with the direct I/O method.

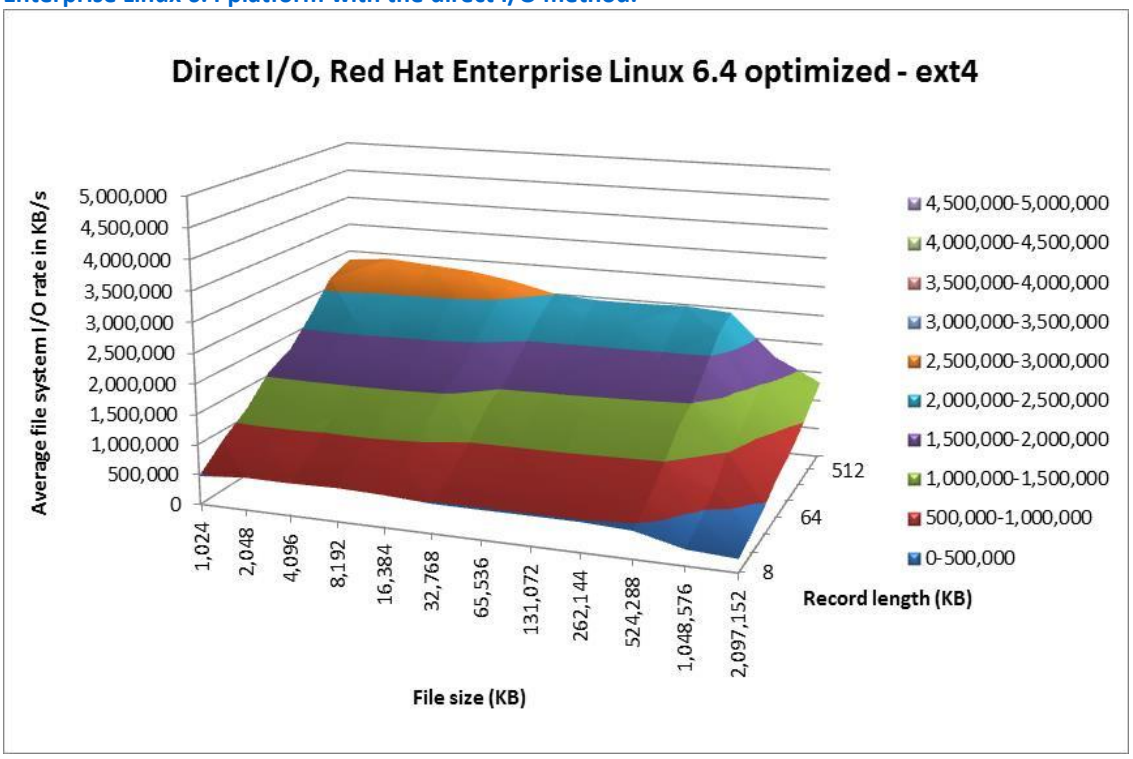

Figure 16: Average IOzone performance in KB/s for ext4 file system on optimized Red Hat Enterprise Linux 6.4 platform with the direct I/O method.

NTFS

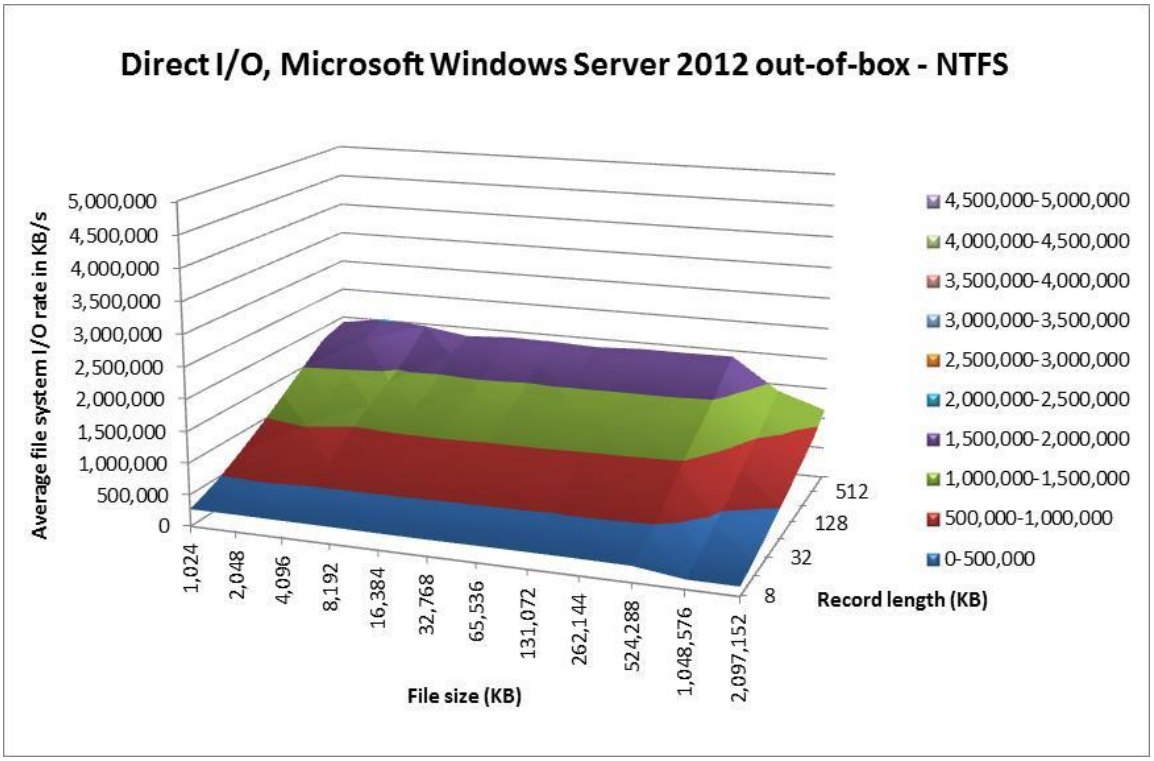

Figure 17: Average IOzone performance in KB/s for NTFS file system on out-of-box Microsoft Windows Server 2012 with the direct I/O method.

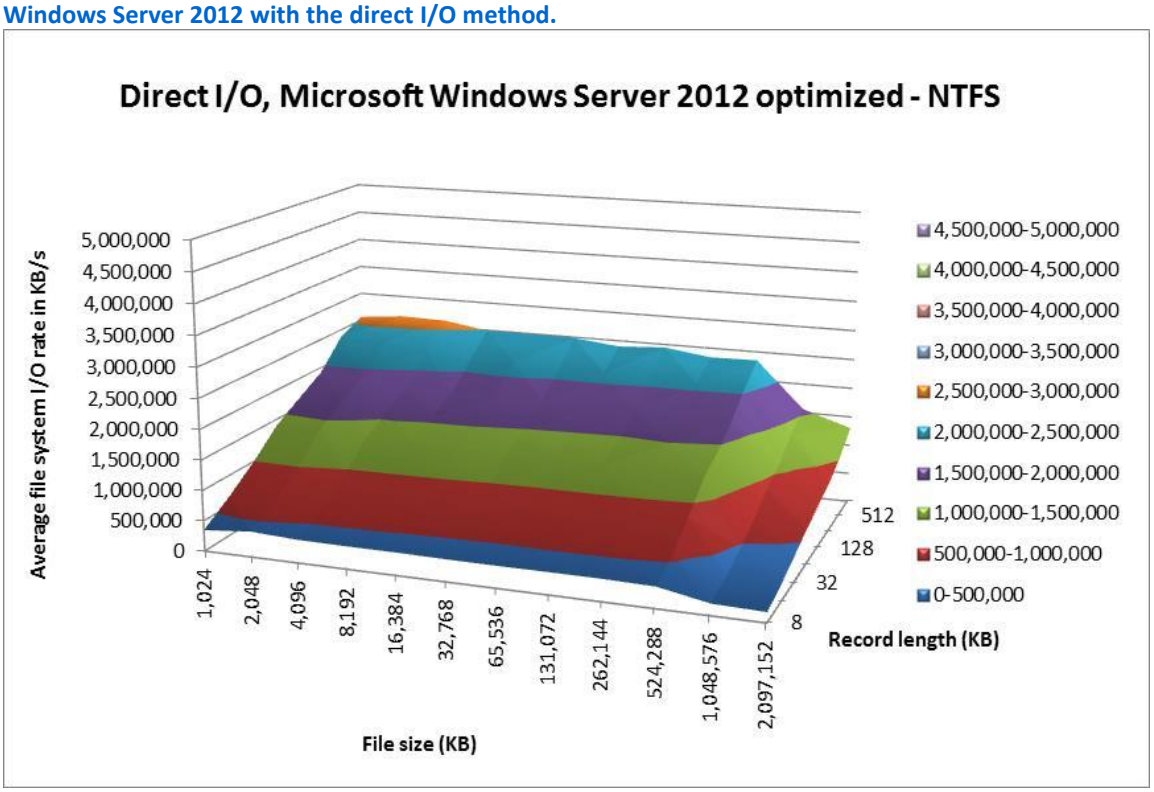

Figure 18: Average IOzone performance in KB/s for NTFS file system on optimized Microsoft Windows Server 2012 with the direct I/O method.

XFS

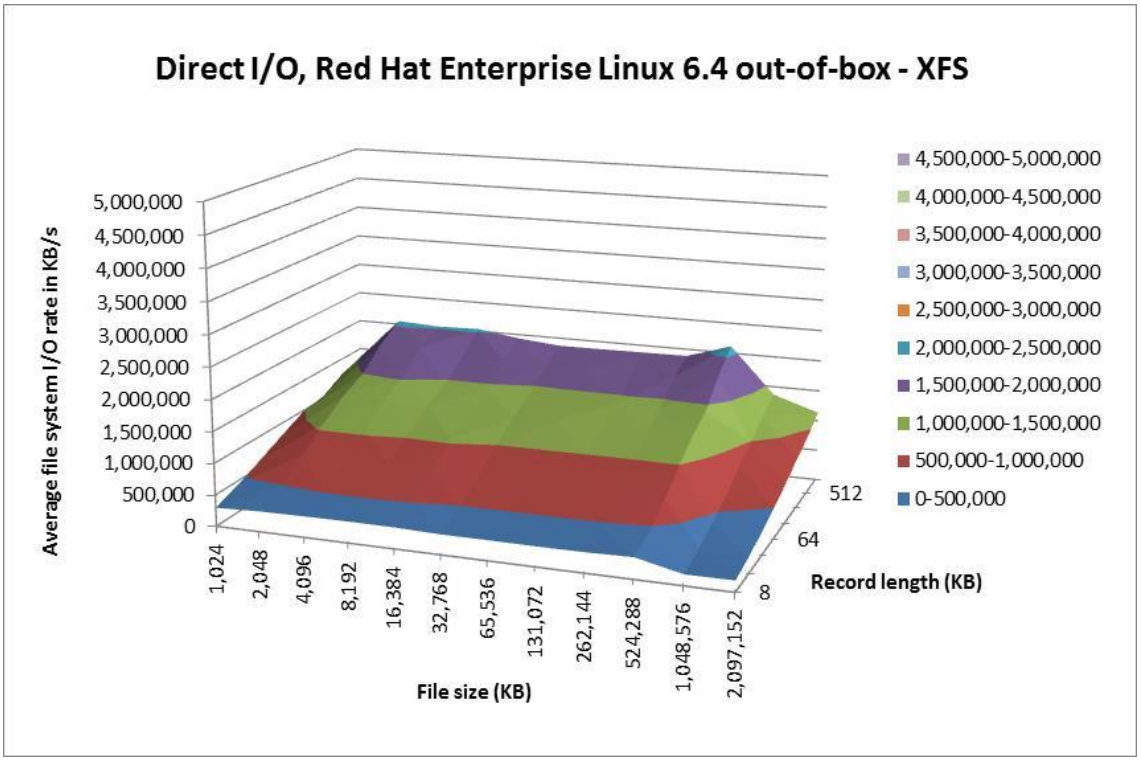

Figure 19: Average IOzone performance in KB/s for XFS file system on out-of-box Red Hat Enterprise Linux 6.4 platform with the direct I/O method.

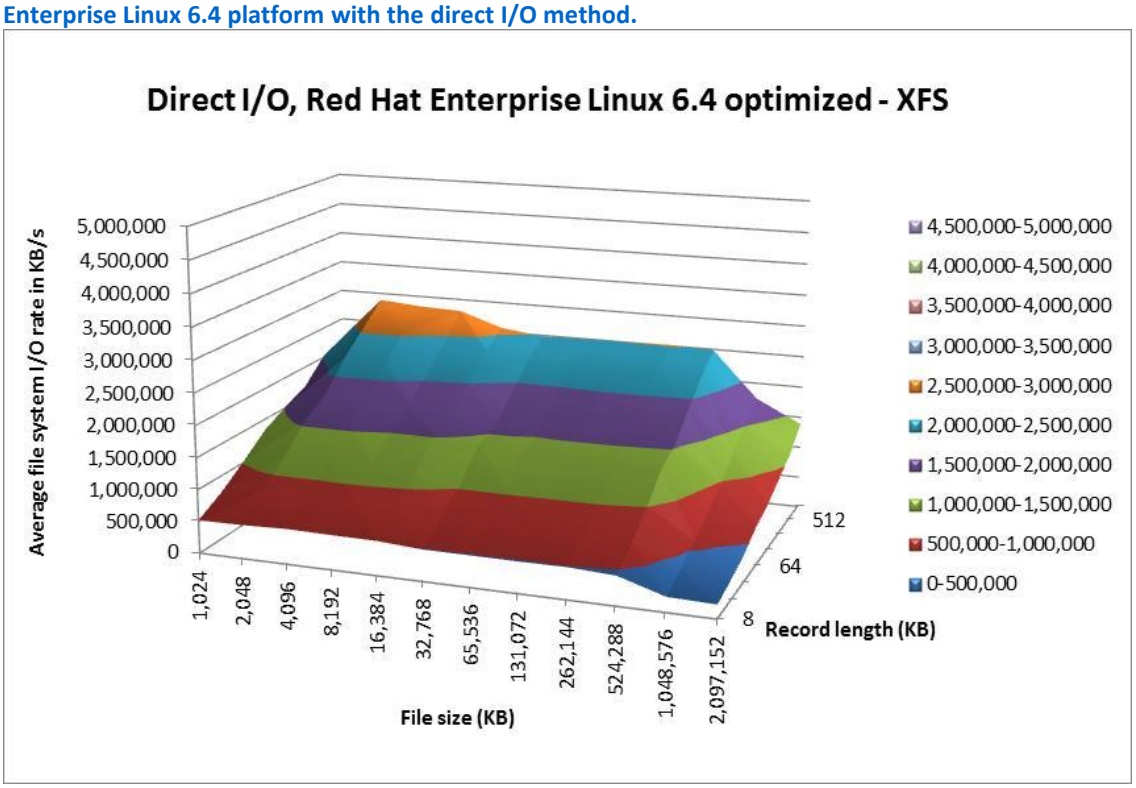

Figure 20: Average IOzone performance in KB/s for XFS file system on optimized Red Hat Enterprise Linux 6.4 platform with the direct I/O method.

ReFS

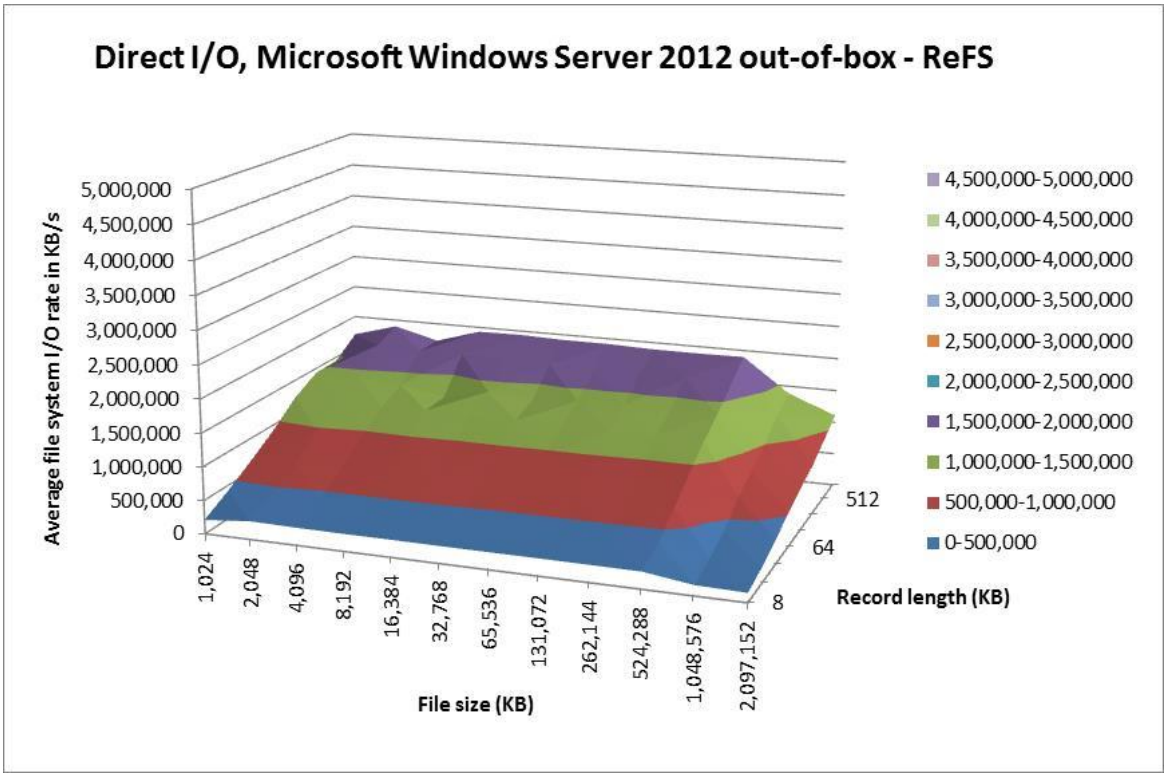

Figure 21: Average IOzone performance in KB/s for ReFS file system on out-of-box Microsoft Windows Server 2012 with the direct I/O method.

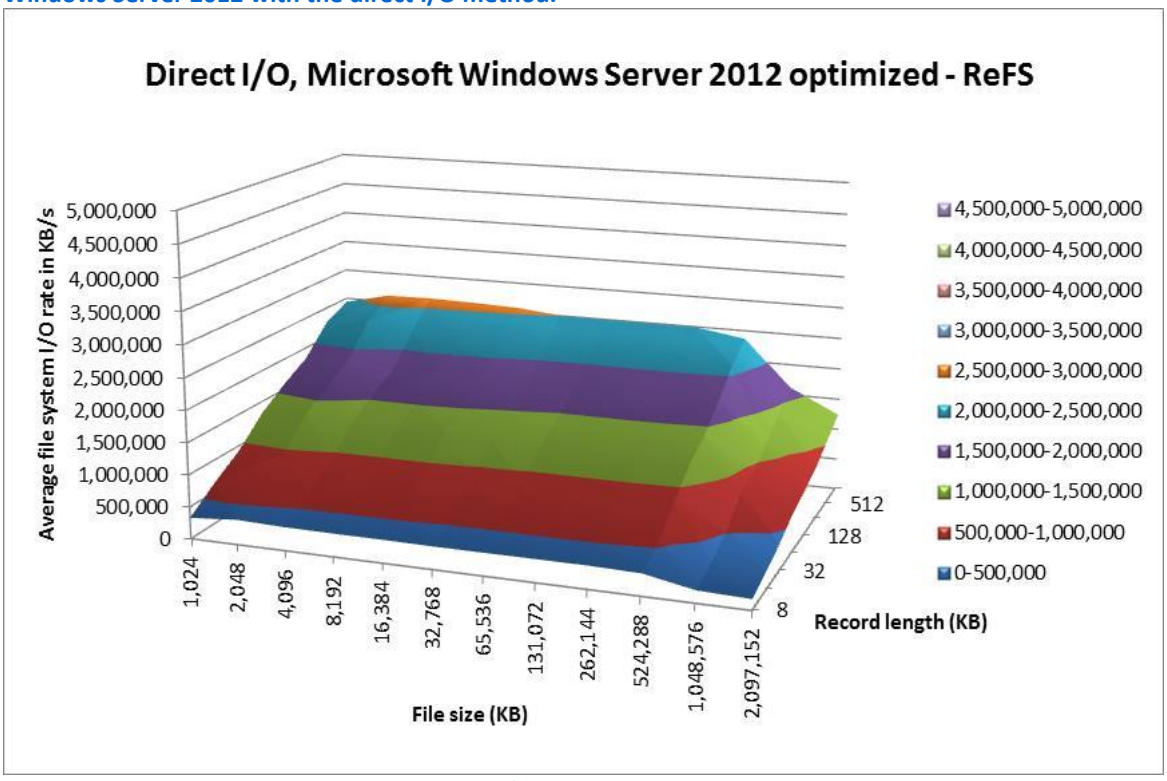

Figure 22: Average IOzone performance in KB/s for ReFS file system on optimized Microsoft Windows Server 2012 with the direct I/O method.

APPENDIX D – DETAILED IOZONE RESULTS

Figures 23 through 28 capture the data points collected during testing with the IOzone benchmark.

In-cache Red Hat Enterprise Linux 6 results

Average file system I/O performance (KB/s): Red Hat Enterprise Linux 6.4 – ext4 – out-of-box configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	2,619,762	2,828,500	3,061,296	3,207,799	3,128,748	2,787,878	2,822,700	2,897,544
	2,048	2,698,331	2,862,802	3,100,792	3,238,848	3,171,723	2,937,001	2,821,115	2,898,712
	4,096	2,730,762	2,885,804	3,128,024	3,232,655	3,149,631	2,972,046	2,845,601	2,866,031
	8,192	2,739,718	2,897,161	3,130,041	3,415,342	3,710,231	3,461,805	3,336,918	3,357,995
	16,384	3,103,121	3,869,641	4,313,797	4,502,324	4,398,668	4,226,442	4,821,675	4,757,327
	32,768	3,632,684	4,360,500	4,826,727	5,011,028	5,379,169	5,122,433	5,141,835	5,309,518
	65,536	4,596,942	4,891,813	5,217,628	5,400,382	5,439,456	5,128,773	5,189,895	5,257,075
	131,072	4,385,187	4,776,719	4,953,052	5,142,409	5,134,675	4,855,048	5,009,551	4,803,365
	262,144	4,197,005	4,453,046	4,739,684	4,839,205	4,917,040	4,654,740	4,854,185	4,742,712
	524,288	3,441,483	4,128,749	3,757,878	3,397,205	3,947,152	4,079,656	4,994,734	4,010,619
	1,048,576	3,927,547	4,485,150	4,920,417	4,882,219	4,623,857	4,698,165	5,029,317	4,872,815
	2,097,152	4,106,970	3,033,503	3,934,227	4,680,614	4,592,325	4,121,093	4,238,161	4,409,589

Average file system I/O performance (KB/s): Red Hat Enterprise Linux 6.4 – ext4 – optimized configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	6,073,374	6,896,276	7,397,456	7,904,428	7,968,753	7,052,057	7,190,163	7,418,309
	2,048	6,503,104	6,995,450	7,644,887	8,045,070	8,094,666	7,474,658	7,161,534	7,320,373
	4,096	6,629,261	7,150,700	7,732,236	8,138,114	7,997,665	7,514,423	7,183,217	7,255,492
	8,192	6,683,646	7,200,623	7,817,523	8,210,626	8,072,850	7,475,986	7,180,105	7,236,401
	16,384	6,450,105	7,000,206	7,544,456	7,973,507	5,193,125	7,198,868	6,888,356	6,839,669
	32,768	4,724,013	5,049,240	5,363,476	5,547,585	5,596,335	5,279,246	5,314,708	5,361,939
	65,536	4,589,370	4,926,612	5,249,255	5,455,440	5,490,065	5,172,193	5,228,551	5,282,607
	131,072	4,581,152	4,918,813	5,242,274	5,451,704	5,491,907	5,176,980	5,231,268	5,294,819
	262,144	4,571,507	4,913,780	5,255,550	5,458,953	5,504,495	5,184,406	5,245,380	5,299,500
	524,288	4,557,504	4,898,999	5,248,440	5,455,295	5,499,620	5,188,866	5,245,817	5,306,438
	1,048,576	4,553,319	4,893,819	5,224,688	5,437,099	5,469,821	5,151,737	5,226,456	5,284,514
	2,097,152	3,634,278	4,777,427	5,117,580	5,208,561	5,202,037	4,934,506	5,009,920	4,405,195

Average file system I/O performance (KB/s): Red Hat Enterprise Linux 6.4 – XFS – out-of-box configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	2,627,659	2,936,059	3,254,553	3,444,610	3,413,419	3,003,990	3,043,737	3,157,848
	2,048	2,709,552	2,972,268	3,346,193	3,511,552	3,498,245	3,220,588	3,078,921	3,127,877
	4,096	2,750,699	3,030,390	3,353,917	3,561,134	3,383,927	3,218,069	3,079,086	3,142,354
	8,192	2,698,471	3,047,803	3,388,898	3,598,899	3,484,056	3,262,918	3,061,449	3,121,469
	16,384	2,661,040	2,909,820	3,248,815	3,442,828	3,348,330	3,137,687	3,001,235	3,008,405
	32,768	2,288,959	2,466,991	2,704,208	2,860,963	2,842,894	2,654,385	2,625,703	2,636,593
	65,536	2,242,591	2,427,365	2,658,211	2,803,888	2,803,830	2,610,480	2,589,380	2,607,665
	131,072	2,239,860	2,424,445	2,656,170	2,800,800	2,803,657	2,615,297	2,585,633	2,610,245
	262,144	2,315,301	2,537,703	2,693,113	2,863,735	2,870,235	2,615,171	2,722,824	2,656,997
	524,288	2,500,321	3,162,553	2,958,214	3,589,496	3,070,618	2,861,692	2,926,368	2,947,608
	1,048,576	2,798,731	2,981,478	3,216,591	3,414,653	3,463,802	3,250,106	3,154,138	3,179,515
	2,097,152	3,054,072	3,340,430	3,928,311	3,908,645	4,526,866	3,825,152	3,615,879	3,819,350

Average file system I/O performance (KB/s): Red Hat Enterprise Linux 6.4 – XFS – optimized configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	6,894,229	7,609,383	8,346,590	8,872,680	8,783,139	7,777,137	7,747,713	8,231,003
	2,048	7,277,527	7,951,180	8,557,760	9,004,370	8,936,395	8,232,754	7,769,691	7,940,687
	4,096	7,399,519	7,975,630	8,695,637	9,120,771	8,857,964	8,338,569	7,904,207	7,943,757
	8,192	7,454,552	8,029,826	8,736,633	9,219,883	8,938,565	8,328,083	7,909,026	7,958,571
	16,384	7,042,377	7,703,105	8,361,397	8,821,951	8,568,075	7,935,632	7,493,623	7,468,151
	32,768	5,075,499	5,417,237	5,810,705	6,045,661	6,057,365	5,712,798	5,731,216	5,775,387
	65,536	4,925,375	5,292,720	5,664,180	5,909,181	5,937,802	5,584,044	5,646,050	5,696,398
	131,072	4,917,198	5,289,875	5,651,045	5,904,577	5,936,561	5,597,569	5,652,272	5,713,380
	262,144	4,914,918	5,214,477	5,663,164	5,921,750	5,958,738	5,605,623	5,658,736	5,720,996
	524,288	4,891,890	5,269,059	5,661,248	5,907,899	5,961,106	5,606,512	5,665,272	5,725,882
	1,048,576	4,883,221	5,275,948	5,664,587	5,923,288	5,963,645	5,620,081	5,673,507	5,722,399
	2,097,152	4,750,906	5,182,162	5,551,678	5,720,931	5,765,372	5,493,944	4,523,860	3,826,407

Figure 23: In-cache method Red Hat Enterprise Linux 6.4 results.

In-cache Microsoft Windows Server 2012 results

Average file system I/O performance (KB/s): Microsoft Windows Server 2012 – NTFS – out-of-box configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	1,792,852	2,354,955	2,981,094	3,447,458	3,536,929	2,958,171	3,021,619	3,062,131
	2,048	1,842,779	2,450,274	3,184,000	3,695,163	3,771,697	3,183,312	3,227,021	3,256,319
	4,096	1,346,225	1,754,812	2,214,080	2,499,250	2,479,837	2,425,935	2,472,293	2,505,025
	8,192	1,268,611	1,671,353	2,117,366	2,438,720	2,446,231	2,227,608	2,406,767	2,426,854
	16,384	1,242,113	1,673,161	2,198,855	2,548,347	2,523,447	2,304,258	2,255,173	2,313,931
	32,768	1,168,710	1,615,827	2,089,637	2,417,658	2,391,101	2,167,542	2,241,967	2,399,971
	65,536	1,545,389	2,167,507	2,646,514	3,102,498	2,058,172	2,051,566	2,272,435	2,478,700
	131,072	1,618,121	2,441,753	2,063,013	2,375,385	2,529,313	2,380,312	2,541,489	2,688,524
	262,144	2,070,968	2,119,190	2,969,178	2,260,377	2,335,837	2,397,729	2,307,477	2,592,580
	524,288	1,810,489	2,305,142	2,278,021	2,562,890	2,475,048	2,371,557	2,406,965	2,531,501
	1,048,576	1,930,289	2,375,990	2,792,338	2,815,162	3,078,204	2,634,531	3,025,452	2,889,683
	2,097,152	2,028,950	2,612,304	3,064,705	3,273,432	3,372,390	3,105,508	3,213,169	3,215,387
Average file system I/O performance (KB/s): Microsoft Windows Server 2012 – NTFS – optimized configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	4,013,978	5,228,365	6,680,711	7,813,329	7,809,804	6,634,143	6,599,714	6,833,208
	2,048	4,144,652	5,463,969	7,114,097	8,250,300	8,435,165	7,011,427	7,290,795	7,311,346
	4,096	3,149,342	4,079,471	5,153,366	5,745,386	5,639,907	5,473,348	5,517,584	5,705,861
	8,192	3,019,841	3,912,891	4,852,373	5,687,295	5,680,765	5,154,998	5,571,796	5,634,339
	16,384	2,985,830	3,816,097	4,860,598	5,585,904	5,555,485	5,099,543	5,147,158	5,463,707
	32,768	2,697,343	3,458,159	4,204,594	4,653,113	4,617,798	4,262,031	4,336,031	4,377,437
	65,536	2,624,745	3,335,759	4,046,737	4,534,647	4,482,453	4,139,708	4,164,708	4,268,033
	131,072	2,625,858	3,384,025	4,080,052	4,506,935	4,439,377	4,101,846	4,161,069	4,222,094
	262,144	2,610,544	3,344,412	4,017,496	4,437,431	4,387,247	4,050,821	4,106,030	4,189,896
	524,288	2,595,568	3,329,041	4,007,432	4,254,092	4,336,243	3,982,758	4,027,647	4,099,736
	1,048,576	2,532,669	3,272,645	3,935,451	4,267,727	4,230,736	3,912,158	3,957,748	4,023,183
	2,097,152	2,505,945	3,214,715	3,902,259	4,237,436	4,197,059	3,888,013	3,950,374	4,032,872

Average file system I/O performance (KB/s): Microsoft Windows Server 2012 – ReFS – out-of-box configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	1,648,444	2,262,030	2,808,068	3,287,867	3,349,695	2,834,876	2,884,157	2,934,526
	2,048	1,748,217	2,340,646	3,042,386	3,506,340	3,634,497	3,019,135	3,130,379	3,106,551
	4,096	1,275,446	1,651,776	2,132,281	2,434,207	2,465,206	2,389,046	2,411,687	2,148,175
	8,192	1,229,972	1,628,887	2,054,318	2,352,845	2,394,604	2,153,367	2,332,453	2,364,813
	16,384	1,183,417	1,569,168	1,942,905	2,230,250	2,220,860	1,948,825	2,039,502	2,171,698
	32,768	1,123,150	1,583,475	2,110,962	2,026,513	2,046,840	1,844,910	1,880,930	1,919,219
	65,536	1,132,699	1,478,227	1,792,349	2,217,501	2,306,738	2,099,955	2,310,379	2,435,434
	131,072	1,526,147	1,978,932	2,089,244	2,380,259	2,593,005	2,470,370	2,642,382	2,792,146
	262,144	1,954,370	1,785,291	2,124,159	2,286,519	2,591,271	2,028,833	2,094,604	2,121,821
	524,288	1,705,004	2,031,925	2,513,640	2,544,075	2,473,459	2,333,523	2,413,096	2,441,778
	1,048,576	1,997,691	2,316,961	2,862,612	3,072,066	3,079,429	2,917,701	2,973,960	2,797,193
	2,097,152	2,078,650	2,607,947	2,980,535	3,361,226	3,368,246	3,096,327	3,211,261	3,195,806

Average file system I/O performance (KB/s): Microsoft Windows Server 2012 – ReFS – optimized configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	4,222,914	4,808,740	5,716,465	6,780,446	6,773,464	6,043,913	6,391,880	6,538,169
	2,048	3,957,970	5,183,477	6,553,777	7,512,688	7,849,510	6,703,031	6,879,048	6,908,219
	4,096	3,142,067	3,929,176	4,752,277	5,451,741	5,436,826	5,293,640	5,338,682	4,873,969
	8,192	3,028,761	3,991,608	5,053,104	5,735,416	5,778,787	5,035,891	5,529,377	5,721,428
	16,384	3,031,241	3,926,451	4,816,171	5,522,518	5,347,306	5,051,777	5,095,231	5,248,450
	32,768	2,753,220	3,506,738	4,195,830	4,662,842	4,623,659	4,270,584	4,333,270	4,395,203
	65,536	2,688,312	3,399,061	4,094,293	4,430,568	4,469,386	4,115,599	4,158,683	4,243,249
	131,072	2,679,838	3,401,410	4,070,976	4,455,517	4,433,542	4,023,313	4,108,097	4,201,726
	262,144	2,667,838	3,356,256	4,019,609	4,395,564	4,373,818	4,023,834	4,163,744	4,212,642
	524,288	2,655,207	3,366,332	3,971,930	4,333,024	4,320,806	3,971,155	3,983,135	4,072,109
	1,048,576	2,585,403	3,260,737	3,894,735	4,225,453	4,199,297	3,897,600	3,933,349	3,999,379
	2,097,152	2,564,502	3,243,937	3,859,049	4,170,794	4,169,853	3,876,843	3,904,150	3,983,933

Figure 24: In-cache method Microsoft Windows Server 2012 results.

Direct I/O Red Hat Enterprise Linux 6 results

Average file system I/O performance (KB/s): Red Hat Enterprise Linux 6.4 – ext4 – out-of-box configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	303,895	543,265	805,574	1,097,618	1,351,531	1,496,485	1,815,363	2,011,223
	2,048	346,252	547,750	808,876	1,105,256	1,356,103	1,548,026	1,811,372	2,065,572
	4,096	344,283	540,974	800,679	1,107,505	1,359,724	1,552,315	1,637,172	2,029,477
	8,192	339,582	539,474	797,313	1,093,129	1,349,384	1,544,927	1,831,745	2,014,393
	16,384	337,149	520,050	783,519	1,076,246	1,322,220	1,509,558	1,799,514	1,968,268
	32,768	321,847	501,362	744,825	1,009,889	1,268,977	1,445,567	1,738,836	1,910,139
	65,536	320,286	491,736	733,826	1,010,757	1,262,181	1,440,263	1,730,261	1,874,335
	131,072	319,911	497,973	724,834	983,693	1,272,368	1,412,763	1,761,472	1,930,475
	262,144	341,946	563,194	844,545	1,145,268	1,418,634	1,613,978	1,960,267	2,019,426
	524,288	349,531	557,450	840,104	1,038,690	1,422,059	1,683,229	1,795,451	2,232,687
	1,048,576	208,293	311,021	491,235	733,841	883,906	1,019,867	1,394,721	1,467,706
	2,097,152	192,176	287,409	386,891	527,556	678,249	803,643	971,999	1,168,732

Average file system I/O performance (KB/s): Red Hat Enterprise Linux 6.4 – ext4 – optimized configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	475,202	843,878	1,134,158	1,553,289	1,753,928	2,180,351	2,660,613	2,840,028
	2,048	540,919	845,736	1,172,427	1,571,947	1,782,157	2,220,343	2,603,109	2,910,103
	4,096	535,966	803,523	1,175,031	1,573,187	1,776,211	2,180,032	2,575,565	2,854,237
	8,192	554,081	828,704	1,178,559	1,565,828	1,765,329	2,168,402	2,561,671	2,799,062
	16,384	531,084	801,577	1,139,390	1,529,800	1,718,213	2,101,661	2,476,212	2,679,584
	32,768	485,821	720,949	1,032,642	1,391,979	1,571,407	1,925,097	2,282,068	2,506,240
	65,536	483,462	721,214	1,030,069	1,374,617	1,563,197	1,916,026	2,262,945	2,450,311
	131,072	485,309	724,074	1,029,873	1,380,297	1,548,147	1,912,585	2,269,814	2,452,984
	262,144	484,434	712,750	1,012,586	1,361,230	1,535,552	1,898,794	2,227,673	2,476,068
	524,288	448,906	684,619	992,053	1,339,015	1,514,342	1,877,317	2,249,080	2,421,970
	1,048,576	259,633	423,421	643,614	1,012,704	1,124,771	1,309,681	1,578,753	1,703,847
	2,097,152	214,977	329,235	469,226	657,372	758,183	919,336	1,132,573	1,319,934

Average file system I/O performance (KB/s): Red Hat Enterprise Linux 6.4 – XFS – out-of-box configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	297,572	443,851	621,686	804,952	983,510	1,043,346	1,236,789	1,305,144
	2,048	324,243	497,000	781,162	1,060,978	1,320,791	1,499,064	1,827,558	2,054,685
	4,096	333,629	523,417	775,580	1,066,819	1,329,289	1,536,497	1,837,075	2,017,721
	8,192	337,490	531,471	778,183	1,029,061	1,283,311	1,457,068	1,715,793	2,035,906
	16,384	335,021	520,586	778,627	1,047,575	1,248,427	1,456,366	1,734,406	1,933,290
	32,768	315,704	490,711	722,245	987,800	1,236,593	1,396,170	1,701,550	1,867,836
	65,536	316,671	488,577	717,165	985,036	1,259,701	1,416,900	1,706,121	1,871,512
	131,072	313,346	490,238	719,482	984,852	1,238,464	1,426,515	1,697,750	1,879,887
	262,144	316,560	486,377	715,455	984,381	1,239,295	1,406,531	1,733,257	1,887,488
	524,288	333,040	459,990	704,551	990,562	1,252,749	1,435,866	1,819,163	2,114,392
	1,048,576	170,661	284,267	436,810	589,454	811,367	983,013	1,213,340	1,365,008
	2,097,152	176,170	261,591	365,718	499,179	630,555	795,465	938,309	1,135,592

Average file system I/O performance (KB/s): Red Hat Enterprise Linux 6.4 – XFS – optimized configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	523,597	741,828	1,019,974	1,324,163	1,514,435	1,673,705	2,031,968	2,169,995
	2,048	532,136	858,443	1,196,265	1,573,582	1,801,637	2,227,982	2,551,991	2,886,363
	4,096	554,385	859,868	1,206,310	1,574,859	1,790,510	2,212,476	2,535,887	2,829,449
	8,192	544,130	882,053	1,202,326	1,511,574	1,736,514	2,066,139	2,396,988	2,801,815
	16,384	538,799	843,325	1,181,721	1,514,548	1,670,690	2,037,496	2,345,647	2,570,246
	32,768	492,551	758,964	1,050,546	1,428,162	1,557,322	1,885,119	2,248,079	2,482,157
	65,536	491,219	765,444	1,058,314	1,385,047	1,552,793	1,898,583	2,249,211	2,492,487
	131,072	487,699	768,539	1,051,368	1,380,174	1,570,429	1,929,290	2,272,498	2,505,801
	262,144	492,711	763,075	1,053,547	1,382,992	1,563,211	1,920,697	2,282,693	2,515,937
	524,288	456,611	715,725	1,007,396	1,359,533	1,545,452	1,897,716	2,237,353	2,481,974
	1,048,576	233,930	388,366	560,862	770,253	976,167	1,246,239	1,530,283	1,744,789
	2,097,152	214,568	305,861	420,702	607,307	723,774	906,042	1,085,448	1,387,700

Figure 25: Direct I/O method Red Hat Enterprise Linux 6.4 results.

Direct I/O Microsoft Windows Server 2012 results

Average file system I/O performance (KB/s): Microsoft Windows Server 2012 – NTFS – out-of-box configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	276,288	441,252	678,156	936,186	1,222,384	1,515,347	1,798,947	1,909,831
	2,048	276,693	466,296	691,059	1,022,902	1,199,358	1,457,823	1,866,210	2,022,013
	4,096	272,239	440,326	677,297	935,543	1,192,575	1,384,978	1,635,426	1,962,086
	8,192	269,210	438,605	684,327	970,793	1,202,397	1,355,045	1,715,860	1,852,850
	16,384	263,100	435,695	684,075	980,123	1,229,111	1,407,840	1,737,505	1,896,309
	32,768	260,306	430,001	673,889	964,991	1,206,701	1,377,124	1,691,041	1,886,161
	65,536	261,906	428,231	666,286	964,218	1,219,926	1,414,257	1,672,998	1,858,324
	131,072	261,906	424,679	645,177	943,261	1,223,837	1,405,536	1,675,125	1,893,991
	262,144	258,288	425,549	666,615	942,164	1,202,082	1,408,648	1,669,008	1,891,638
	524,288	250,654	411,668	651,381	926,209	1,166,623	1,372,986	1,653,065	1,901,663
	1,048,576	149,495	266,617	444,799	558,295	786,562	961,391	1,162,407	1,390,765
	2,097,152	145,893	240,476	358,428	495,351	620,160	761,675	905,924	1,143,149

Average file system I/O performance (KB/s): Microsoft Windows Server 2012 – NTFS – optimized configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	349,953	651,783	995,906	1,397,945	1,659,039	1,913,797	2,377,858	2,560,442
	2,048	421,368	659,214	1,013,569	1,436,138	1,697,316	1,914,048	2,341,796	2,629,932
	4,096	381,388	611,374	946,430	1,336,194	1,578,389	1,800,171	2,256,750	2,615,014
	8,192	374,295	600,017	941,405	1,337,318	1,546,548	1,733,721	2,127,916	2,491,649
	16,384	373,142	582,238	933,209	1,323,566	1,537,561	1,767,638	2,181,887	2,493,111
	32,768	363,049	573,944	906,573	1,287,246	1,503,494	1,744,594	2,100,920	2,484,519
	65,536	359,778	571,271	897,188	1,284,528	1,480,576	1,737,931	2,118,526	2,397,833
	131,072	361,594	579,904	897,403	1,284,197	1,463,411	1,732,707	2,081,468	2,446,783
	262,144	360,160	578,417	892,653	1,266,271	1,495,778	1,712,474	2,080,109	2,338,298
	524,288	343,722	554,142	867,237	1,235,931	1,441,145	1,702,715	2,057,703	2,350,779
	1,048,576	187,775	298,965	471,654	663,977	894,695	1,079,567	1,355,443	1,563,032
	2,097,152	166,536	274,962	413,010	572,061	698,675	825,085	1,027,242	1,300,691

Average file system I/O performance (KB/s): Microsoft Windows Server 2012 – ReFS – out-of-box configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	209,514	426,326	666,838	925,494	1,226,954	1,430,314	1,440,290	1,703,592
	2,048	274,703	448,185	693,340	970,139	1,222,299	1,421,286	1,759,395	1,897,129
	4,096	263,368	429,255	660,472	941,607	1,207,064	1,367,806	1,725,595	1,718,425
	8,192	259,327	429,592	674,619	954,643	1,229,140	1,242,240	1,740,123	1,929,734
	16,384	258,826	432,124	674,767	939,117	1,218,944	1,397,689	1,728,771	1,936,617
	32,768	256,448	426,168	656,506	943,885	1,220,790	1,331,269	1,708,987	1,914,894
	65,536	255,340	421,090	657,092	938,963	1,202,033	1,389,892	1,664,757	1,914,887
	131,072	255,651	418,502	659,051	935,327	1,217,142	1,392,640	1,641,035	1,891,653
	262,144	253,217	418,210	651,973	922,383	1,197,450	1,365,431	1,683,003	1,890,137
	524,288	250,398	407,961	651,933	925,612	1,195,160	1,398,351	1,643,765	1,897,086
	1,048,576	152,722	262,813	418,895	674,423	839,687	994,493	1,245,162	1,384,216
	2,097,152	138,789	237,391	358,691	520,939	643,918	759,165	929,086	1,108,023

Average file system I/O performance (KB/s): Microsoft Windows Server 2012 – ReFS – optimized configuration									
		Record length (KB)							
		8	16	32	64	128	256	512	1,024
File size (KB)	1,024	330,471	647,213	925,074	1,317,549	1,592,241	1,826,885	2,250,366	2,438,461
	2,048	389,642	627,799	974,108	1,383,614	1,650,396	1,897,493	2,355,999	2,606,508
	4,096	367,363	591,464	919,987	1,298,302	1,540,514	1,757,227	2,213,922	2,616,761
	8,192	361,822	588,261	919,173	1,288,611	1,546,544	1,810,322	2,221,230	2,601,961
	16,384	360,490	589,415	923,005	1,310,923	1,528,314	1,805,429	2,190,521	2,574,976
	32,768	353,940	568,792	898,426	1,266,029	1,489,712	1,752,264	2,130,682	2,506,375
	65,536	353,731	568,599	884,791	1,220,287	1,440,635	1,736,894	2,121,661	2,492,902
	131,072	355,377	572,177	892,710	1,225,717	1,445,815	1,756,305	2,129,223	2,495,431
	262,144	355,004	570,684	891,902	1,226,504	1,444,683	1,743,874	2,127,976	2,486,843
	524,288	338,826	547,214	877,468	1,213,881	1,416,684	1,733,293	2,104,887	2,365,806
	1,048,576	189,087	321,397	517,879	831,242	941,110	1,135,437	1,434,280	1,617,478
	2,097,152	166,051	272,257	413,470	591,304	701,791	835,329	1,036,080	1,241,613

Figure 26: Direct I/O method Microsoft Windows Server 2012 results.

Out-of-cache Red Hat Enterprise Linux 6.4 results

Average file system I/O performance (KB/s): Red Hat Enterprise Linux 6.4 – ext4 – out-of-box configuration		
File size (KB)	Record length (KB)	
	64	1,024
33,554,432	498,332	628,617
Average file system I/O performance (KB/s): Red Hat Enterprise Linux 6.4 – ext4 – optimized configuration		
File size (KB)	Record length (KB)	
	64	1,024
33,554,432	619,722	894,352
Average file system I/O performance (KB/s): Red Hat Enterprise Linux 6.4 – XFS – out-of-box configuration		
File size (KB)	Record length (KB)	
	64	1,024
33,554,432	519,110	647,652
Average file system I/O performance (KB/s): Red Hat Enterprise Linux 6.4 – XFS – optimized configuration		
File size (KB)	Record length (KB)	
	64	1,024
33,554,432	599,474	992,649

Figure 27: Out-of-cache method Red Hat Enterprise Linux 6.4 results.

Out-of-cache Microsoft Windows Server 2012 results

Average file system I/O performance (KB/s): Microsoft Windows Server 2012 – NTFS – out-of-box configuration		
File size (KB)	Record length (KB)	
	64	1,024
33,554,432	417,644	651,293
Average file system I/O performance (KB/s): Microsoft Windows Server 2012 – NTFS – optimized configuration		
File size (KB)	Record length (KB)	
	64	1,024
33,554,432	519,317	814,517
Average file system I/O performance (KB/s): Microsoft Windows Server 2012 – ReFS – out-of-box configuration		
File size (KB)	Record length (KB)	
	64	1,024
33,554,432	441,695	683,938
Average file system I/O performance (KB/s): Microsoft Windows Server 2012 – ReFS – optimized configuration		
File size (KB)	Record length (KB)	
	64	1,024
33,554,432	557,495	821,037

Figure 28: Out-of-cache method Microsoft Windows Server 2012 results.

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.