Technical White Paper

Comparison of Novell, Polyserve, and Microsoft's Clustering Performance

J Wolfgang Goerlich
Written December 2006

Business Abstract

Performance is a priority in highly utilized, asynchronous environments. Storage Area Networks (Sans) improve data management and offer ease-of-use. However, the move from dedicated to network disk changes the performance dynamic. Add clustering for high-availability and performance changes even more, possibly getting out of control.

This paper details performance testing performed against three clustering technologies – Novell Netware Clustering Services, PolyServe Matrix Server, and Microsoft Windows Clustering – on a high traffic Compellent San. Each clustering technology is evaluated and a final recommendation is made in the summary.

Table of Contents

Comparison of Novell, Polyserve, and Microsoft's Clustering Performance 1
Business Abstract 1
Table of Contents 3
Section 1: Test Environment 4
Client Computers 4
Server Computers 5
Switches and Cabling 5
Section 2: Testing Methodology 12
Local I/O Testing 12
Client-Server I/O Testing 15
Network I/O Test: Compellent Caching 16
Section 3: Test Results 19
Local I/O Test 1 19
Local I/O Test 2 28
Network I/O Test 32
64K Read/Write 33
Section 4: Summary 35
Where is the bottleneck? 35
Which clustering technology offers the best performance in this environment? 35

Section 1: Test Environment

The test clusters were built on an IBM Blade Center JS21, using three blade servers. These connect to the clients via a Gigabit Ethernet Local Area Network (Lan), and connect to the disks via a 2 Gigabit Fibre Channel Storage Area Network (San). See Figure x.

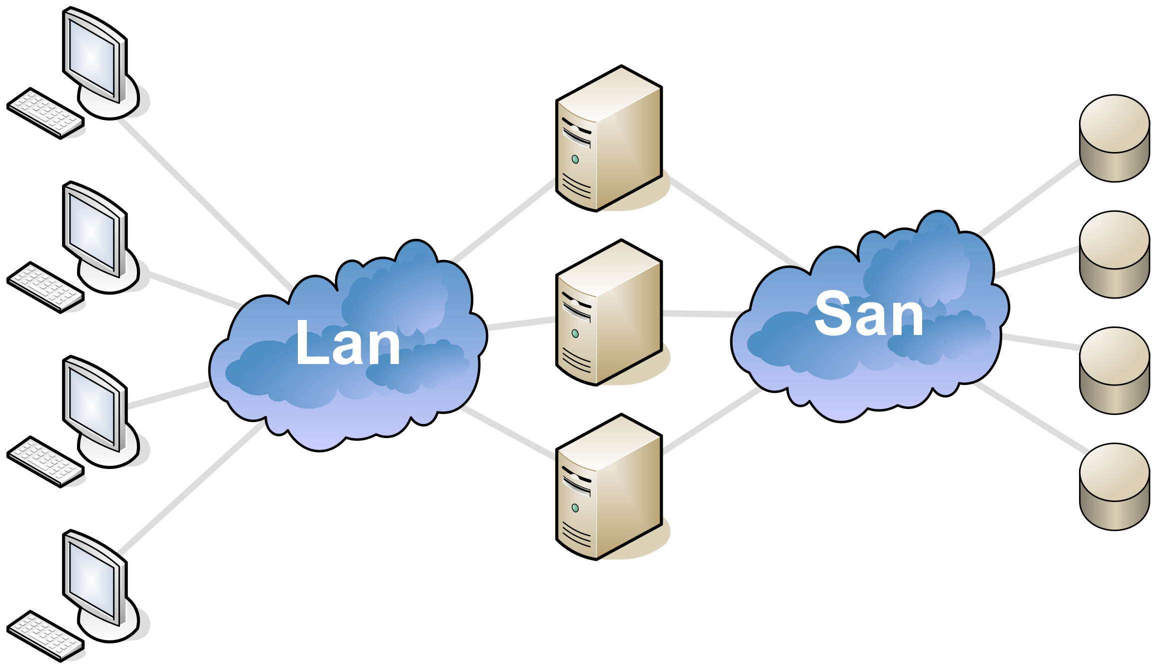


Figure x: Example Test Environment

Client Computers

Processor	(Various)
Memory	2 GB
Network Interface Card	Intel Pro/1000 MT Intel Driver 8.6.12.1, 10/21/2005
Link Speed & Duplex	100 Mbps / Full Duplex
Operating System	Windows XP SP2

Server Computers

Blade Enclosure	IBM Blade Center JS21
Blade Server	IBM HS20 Blade (8842-41U)
Processor	Dual 3.6 GHz, Hyper-threading enabled
Memory	4 GB
Network Interface Card	Broadcom NetXtreme Gigabit
Host Bus Adapter	Dual Qlogic QLA2312
Link Speed & Duplex	1 Gbps / Full Duplex

Switches and Cabling

Each IBM Blade Server has two NICs and two HBAs. IBM Blade Center routes these onto a bus and out two four I/O modules, see figure X. One NIC and two HBAs are connected to the outside networks for the purposes of these tests.

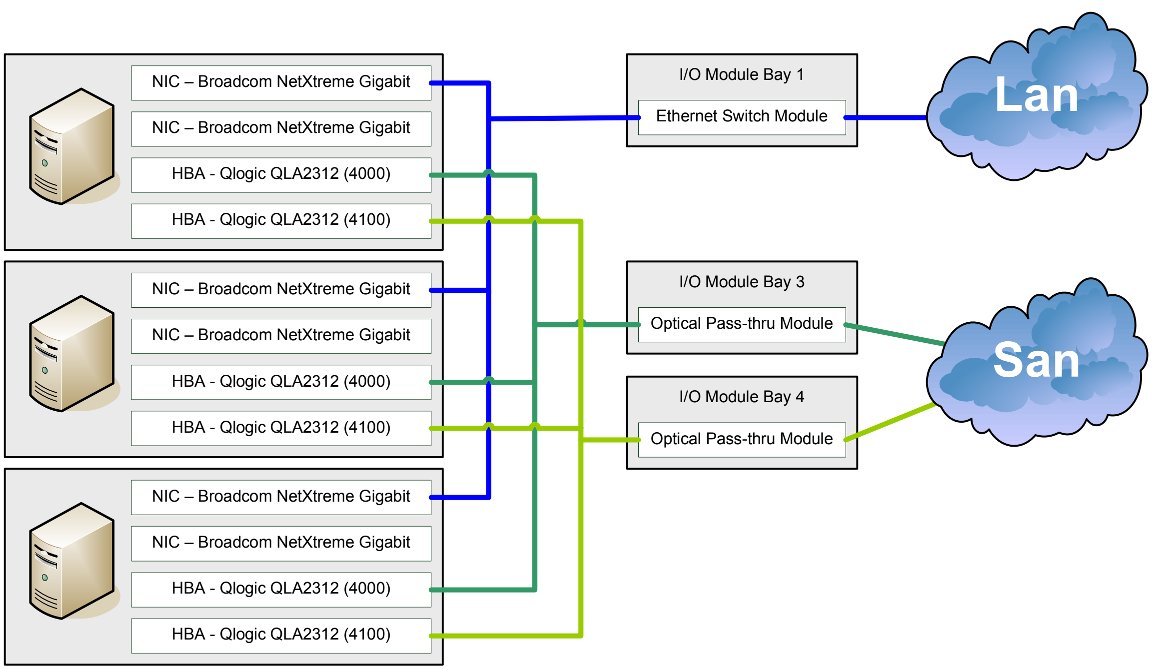


Figure x: Blade Center Bus I/O Architecture

Lan Switch

The blade servers in the Blade Center connect to an internal Ethernet switch module (SM). These connections are dedicated 1 Gb. The Blade Center SM connects to the Lan via six trunked 1 Gb connections to an Extreme Black Diamond 8810.

Lan Switch	Extreme Black Diamond 8810
Firmware	11.3.25

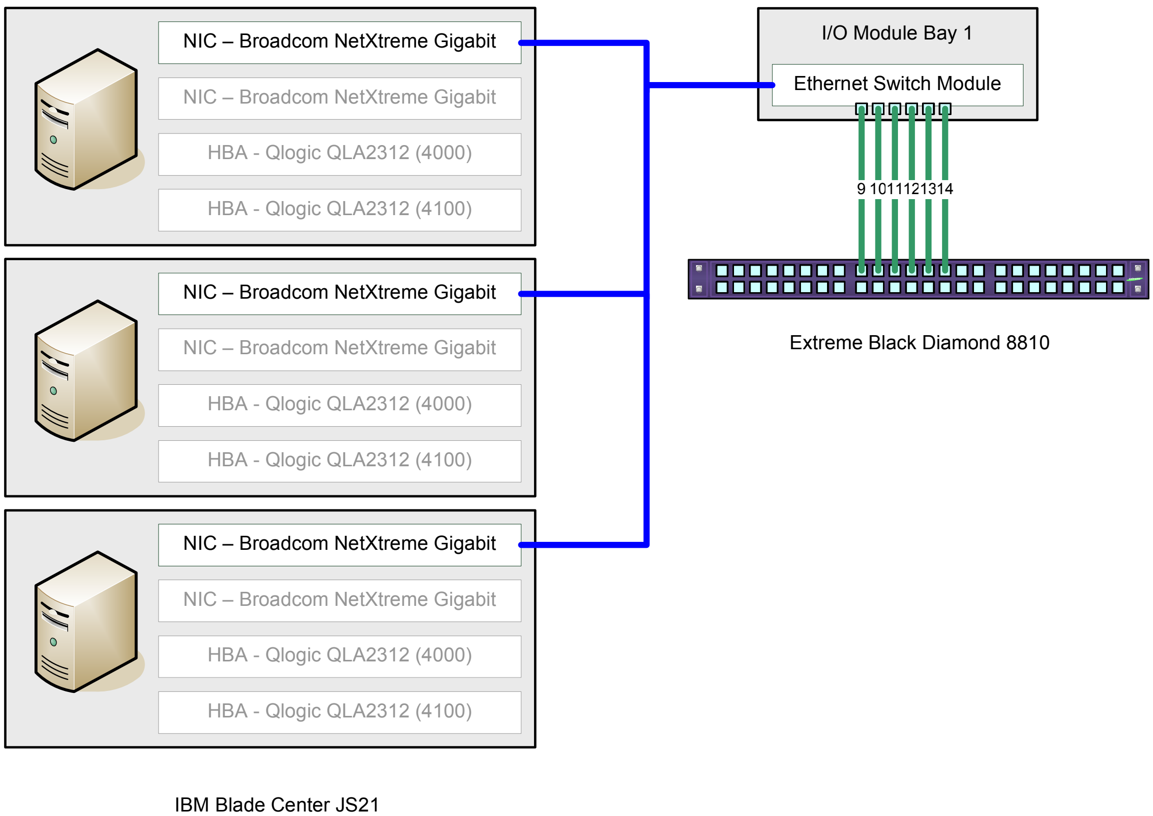


Figure x: Lan Switch and Cabling

Ethernet Card Settings

Network Interface Card	Broadcom NetXtreme Gigabit
Device Driver	Broadcom Driver 9.81.0.0, 8/28/2006
Flow Control	Disabled
Jumbo MTU	1500
Large Send Offload	Enabled
Wake Up Capabilities	None

* This applies to the Polyserve and Microsoft cluster tests only.

Ethernet Cabling

SM Bay	Patch Cable	Junction	Extreme Port
1	48	ATM-18 A	9
2	47	ATM-18 B	10
3	45	ATM-18 C	11
4	42	ATM-18 D	12
5	41	ATM-18 E	13
6	209	ATM-18 F	14

San Switch

San Switch	McData Sphereon 4500
Firmware	08.00.01
Zoning	Port-Level Zoning
Ports Configured for	F-Port, 2 Gig

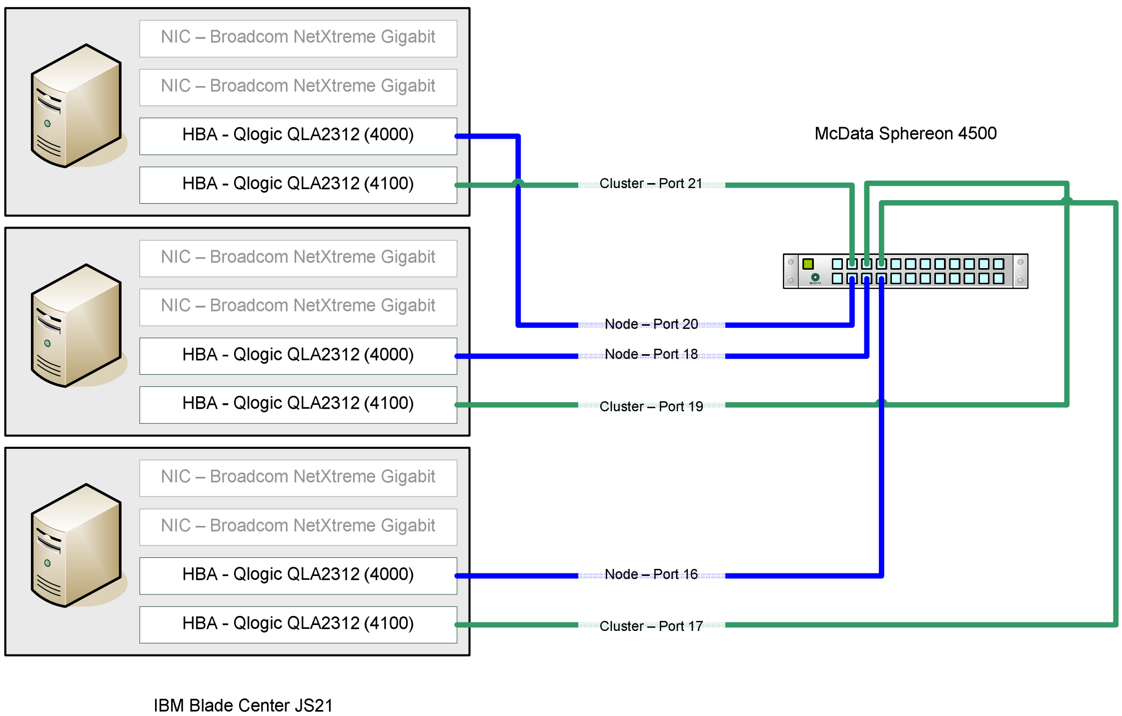


Figure x: San Switch and Cabling

Data Storage

Manufacturer	Compellent
System	Storage Center
Controller	Dual CT-SC010 (Westville)
Firmware	3.3.7
Disks	45

HBA Firmware Settings

Host Adapter Settings	
BIOS Version	1.43
Host Adapter Bios	Enabled
Frame Size	2048
Loop Reset Delay	5
Adapter Hard Loop ID	Disabled
Hard Loop ID	0
Spinup Delay	Enabled
Connection Options	1 (Point to Point Only)
Fibre Channel Tape Support	Disabled
Data Rate	1 (2 Gb/s)
Advanced Adapter Settings	
Execution Throttle	255
Luns per Target	16
Enable LIP Reset	Yes
Enable LIP Full Login	Yes
Enable Target Reset	Yes
Login Retry Count	60
Port Down Retry Count	60
Link Down Timeout	30
Extended Error Logging	Disabled
Operation Mode	0
Interrupt Delay Timer	0

HBA Cabling

Computer	HBA WWN	OPM Bay	Break-out Cable	Extension Cable	Sphereon Nickname	Sphereon Port
Fs6	AF76	3	3-D	MD110	Fs6	20
	AF77	4	3-D	MD111	Fs6 Cluster	21
Fs7	B45E	3	4-A	MD112	Fs7	18
	B45F	4	4-A	MD113	Fs7 Cluster	19
Fs8	B406	3	4-B	MD114	Fs8	16
	B407	4	4-B	MD115	Fs8 Cluster	17

Section 2: Testing Methodology

Tests were ran to measure the performance of the local disk subsystem and of the overall client-server file transfers. To different tools were used for these tests: Iometer and Iozone.

The key variable was the level of Compellent San caching.

Local I/O Testing

The local I/O tests were conducted using Iometer, release 2004.07.30.

http://www.iometer.org/doc/downloads.html

The following tests were performed on the local, clustered disks:

- 2K Streaming Read
- 2K Streaming Write
- 4K Streaming Read
- 4K Streaming Write
- 8K Streaming Read
- 8K Streaming Write
- 32K Streaming Read
- 32K Streaming Write
- 64K Data Streaming Read
- 64K Data Streaming Write
- 128K Data Streaming Read
- 128K Data Streaming Write
- 256K Data Streaming Read
- 256K Data Streaming Write

All three cluster technologies were tested with five levels of Compellent volume caching.

The two Windows-based clustering technologies were also retested to determine the impact of increasing the HBA driver's NumberOfRequest setting.

Both local I/O tests were executed during peak load times; from 8 AM to 12 PM on weekdays. Testing in a production environment means that there is plenty of noise in the data. Tests that had results that were too far out of the bounds (+/- 20%) were re-run.

Local I/O Test 1: Compellent Caching

Cache Level 1: No Caching

[ ] Enable Write Cache
[ ] Enable Read Cache
[ ] Enable Writes to Read Cache
[ ] Enable Read Ahead

Cache Level 2: Write Caching

[x] Enable Write Cache
[ ] Enable Read Cache
[ ] Enable Writes to Read Cache
[ ] Enable Read Ahead

Cache Level 3: Write & Read Caching

[x] Enable Write Cache
[x] Enable Read Cache
[ ] Enable Writes to Read Cache
[ ] Enable Read Ahead

Cache Level 4

[x] Enable Write Cache
[x] Enable Read Cache
[x] Enable Writes to Read Cache
[ ] Enable Read Ahead

Cache Level 5

[x] Enable Write Cache
[x] Enable Read Cache
[x] Enable Writes to Read Cache
[x] Enable Read Ahead

Local I/O Test 2: Storport Driver – NumberofRequests

A second round of testing was added to measure the impact of increased NumberOfRequests.

"This parameter allows you to specify the number of SRBs created for a given adapter. This improves performance and allows Windows to give more disk requests to a logical disk, which is most useful for HW RAID adapters that have concurrency capabilities since each logical disk consists of multiple physical disks. However, the default setting is often less than optimal for many high-speed HW RAID disk arrays. Overall disk array performance can be improved by setting NumberOfRequests to a value in the range of 32 to 96." *(Performance Tuning Guidelines for Windows Server 2003, October 2003)*

Five levels of NumberOfRequests were tested:

NumberOfRequests 1: Null (Defaults to 16)
NumberOfRequests 2: 32
NumberOfRequests 3: 64
NumberOfRequests 4: 128
NumberOfRequests 5: 255

With each test, the number of Iometer's outstanding I/Os were increased accordingly:

16 per target
32 per target
64 per target
128 per target
255 per target

All five tests were executed with full Compellent caching (level 5):

[x] Enable Write Cache
[x] Enable Read Cache
[x] Enable Writes to Read Cache
[x] Enable Read Ahead

These tests were only run on the Polyserve and Microsoft clusters.

Client-Server I/O Testing

The client-server test was designed to simulate users connecting to the clustered file server. Twenty-four test workstations were setup and loaded with the test software. These were connected to the network at 100 Mbps, Full Duplex. Again, the file servers were connected at 1 Gbps Full. All workstations then simultaneously tested the I/O from the servers.

For client-server network I/O, Iozone 3.263 was used. This was done because it has a command line interface. This was necessary for scheduling the test.

http://www.iozone.org/

The command line that was used specified 64K read/writes to a 512 MB file.

iozone.exe -Rab %computername%-out.wks -s 512m -r 64k -e -c -i 0 -i 2 +t

The client-server testing was conducted over on the weekend to minimize the impact on the network. Thus, because the San was not under as great of a load, the available disk I/O is greater in these tests than in the local I/O tests.

The testing was coordinated using Scheduled Tasks. All 24 computers were within one domain and time was synchronized using W32time/NTP. The tests were initiated from a batch file every hour on the hour.

Network I/O Test: Compellent Caching

Cache Level 1: No Caching

[ ] Enable Write Cache
[ ] Enable Read Cache
[ ] Enable Writes to Read Cache
[ ] Enable Read Ahead

Cache Level 2: Write Caching

[x] Enable Write Cache
[ ] Enable Read Cache
[ ] Enable Writes to Read Cache
[ ] Enable Read Ahead

Cache Level 3: Write & Read Caching

[x] Enable Write Cache
[x] Enable Read Cache
[ ] Enable Writes to Read Cache
[ ] Enable Read Ahead

Cache Level 4

[x] Enable Write Cache
[x] Enable Read Cache
[x] Enable Writes to Read Cache
[ ] Enable Read Ahead

Cache Level 5

[x] Enable Write Cache
[x] Enable Read Cache
[x] Enable Writes to Read Cache
[x] Enable Read Ahead

Iotest.bat Batch File

The tests were initiated using a batch file: iotest.bat. This batch mapped a network drive to the test cluster, created a folder in the share, set this as the working directory, and called Iozone.

Having separate working directories were necessary because the name of the temporary file Iozone uses is not configurable. If multiple computers ran tests from the same directory, the first computer would create the temporary file and obtain an oplock. Subsequent computers' tests would then fail.

```
@echo off

rem | iotest.bat -- Client-Server Network I/O Test
rem | Created for the clustering performance tests, uses IOzone v3.263
rem | 2006-12/13 -- J Wolfgang Goerlich
rem |
rem | Parameters:
rem | 1 = the mapping
rem | 2 = the output file name
rem | 3 = the block size
rem |
rem | To run from a Schedule Task:
rem | C:\Cached\iotest.bat \\server\share %computername%-out.wks 64K
rem |
rem | This can be run remotely with Psexec:
rem | psexec \\client -e C:\Cached\iotest.bat
rem |

echo.
echo Clearing existing mappings . . .
Net Use M: /Delete

echo Mapping to the file server . . .
Net Use M: "%1" /Persistent:no

echo Pausing for ten seconds . . .
@ping -n 10 127.0.0.1 > NUL 2>&1
echo Done.
echo.

echo.
echo Creating a new test folder . . .
M:
MD M:\%computername%
CD M:\%computername%
CD
echo.

echo Starting the test . . .
echo on
"C:\Program Files\Benchmarks\Iozone 3.263\iozone.exe" -Rab %2 -s 512m -
r %3 -e -c -i 0 -i 2 +t
@echo off
echo.
echo Done.
```

```
echo.
exit

rem | Sample output:
rem |
rem | C:\>iotest.bat password
rem | Clearing existing mappings . . .
rem | M: was deleted successfully.
rem |
rem | Mapping to the file server . . .
rem | The command completed successfully.
rem |
rem | Pausing for ten seconds . . .
rem | Done.
rem |
rem | Clearing existing test folder . . .
rem | Done.
rem |
rem | Creating a new test folder . . .
rem | M:\MF0
rem |
rem | Running the test . . .
rem |
rem | M:\MF0>"C:\Program Files\Benchmarks\Iozone 3.263\iozone.exe" -Rab
MF0-out.wks -s
rem |  512m -r 64k -e -c -i 0 -i 2 +t
rem |         Iozone: Performance Test of File I/O
rem |                 Version $Revision: 3.263 $
rem |                 Compiled for 32 bit mode.
rem |                 Build: Windows
rem |
rem | Done.
rem |
```

Netware Cluster Test.exe

One wrinkle was that Scheduled Tasks cannot implicitly authenticate to Novell Netware. This is because the task launches as a process, without access to the shell or Gina. We worked around this by creating a simple .Net 2 application that logs onto Netware and then calls the Iotest.bat.

This application utilized the clsNovellAPINET by Karl Durrance (1.08).
http://forge.novell.com/modules/xfmod/project/?vbnet-api-class

Thus, in the Netware tests, the Scheduled Task called “Netware Cluster Test.exe”, which logged in and then launched Iotest.bat.

Section 3: Test Results

Local I/O Test 1

2K Read MBps	1	2	3	4	5
Microsoft	4.56	5.19	5.24	3.89	21.13
Novell	4.06	5.05	7.75	7.73	9.28
Polyserve	3.35	4.69	3.89	11.45	9.14

2K Write MBps	1	2	3	4	5
Microsoft	0.95	1.92	1.07	0.95	1.04
Novell	1.94	1.96	1.81	2.63	0.77
Polyserve	0.16	0.93	0.80	0.99	0.97

4K Read MBps	1	2	3	4	5
Microsoft	8.30	10.20	7.51	7.26	31.28
Novell	7.15	7.03	7.93	8.08	10.13
Polyserve	3.35	8.28	7.43	20.63	17.44

4K Write MBps	1	2	3	4	5
Microsoft	1.90	3.83	1.80	2.14	1.83
Novell	2.15	4.08	1.00	0.91	1.38
Polyserve	0.44	1.79	1.35	1.78	1.85

8K Read MBps	1	2	3	4	5
Microsoft	13.87	17.92	14.35	14.32	42.22
Novell	6.13	8.92	9.5	10.33	11.03
Polyserve	7.38	16.44	10.36	38.41	31.97

8K Write MBps	1	2	3	4	5
Microsoft	3.83	8.23	4.37	4.74	3.76
Novell	0.53	0.58	3.19	2.94	1.54
Polyserve	0.87	3.29	3.03	3.82	4.49

32K Read MBps	1	2	3	4	5
Microsoft	34.15	49.44	44.33	42.63	96.12
Novell	21.75	27.27	21.52	31.48	38.69
Polyserve	14.06	40.80	37.51	66.24	68.79

32K Write MBps	1	2	3	4	5
Microsoft	14.87	29.31	14.29	16.72	15.53
Novell	12.61	8.01	8.95	10.80	16.59
Polyserve	5.11	11.86	11.24	15.12	13.74

64K Read MBps	1	2	3	4	5
Microsoft	75.72	79.95	64.80	85.13	143.83
Novell	38.07	63.79	79.05	82.91	81.04
Polyserve	45.60	74.35	69.36	79.65	80.11

64K Write MBps	1	2	3	4	5
Microsoft	31.05	54.46	25.00	33.57	39.07

Novell	17.32	35.72	45.06	40.21	39.79
Polyserve	8.92	21.56	20.73	25.85	26.76

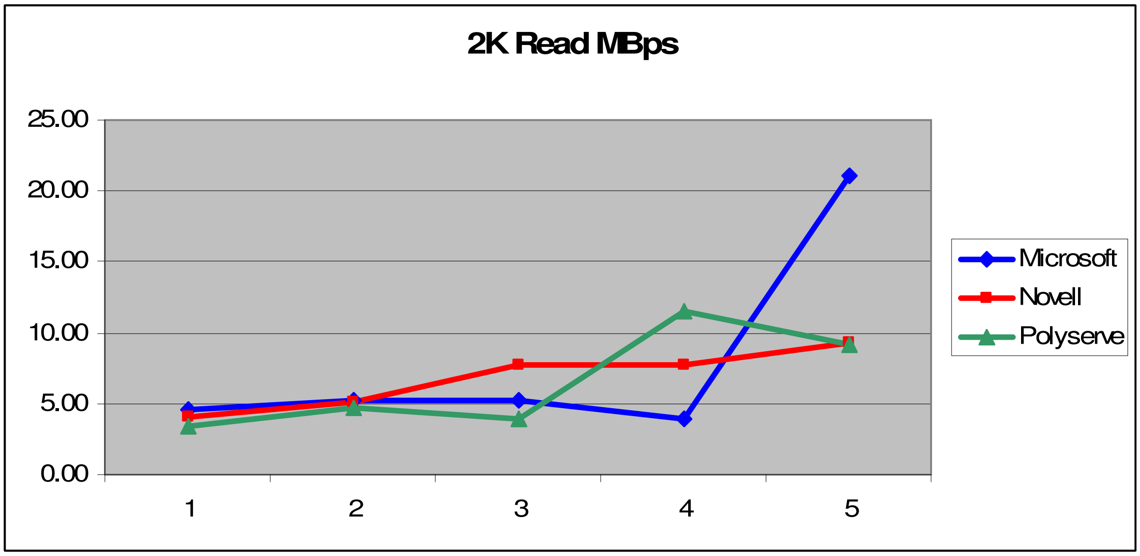
128K Read MBps	**1**	**2**	**3**	**4**	**5**
Microsoft	130.36	133.19	119.83	129.82	144.92
Novell	65.03	66.92	60.8	66.24	74.21
Polyserve	75.24	114.42	109.23	111.76	91.61

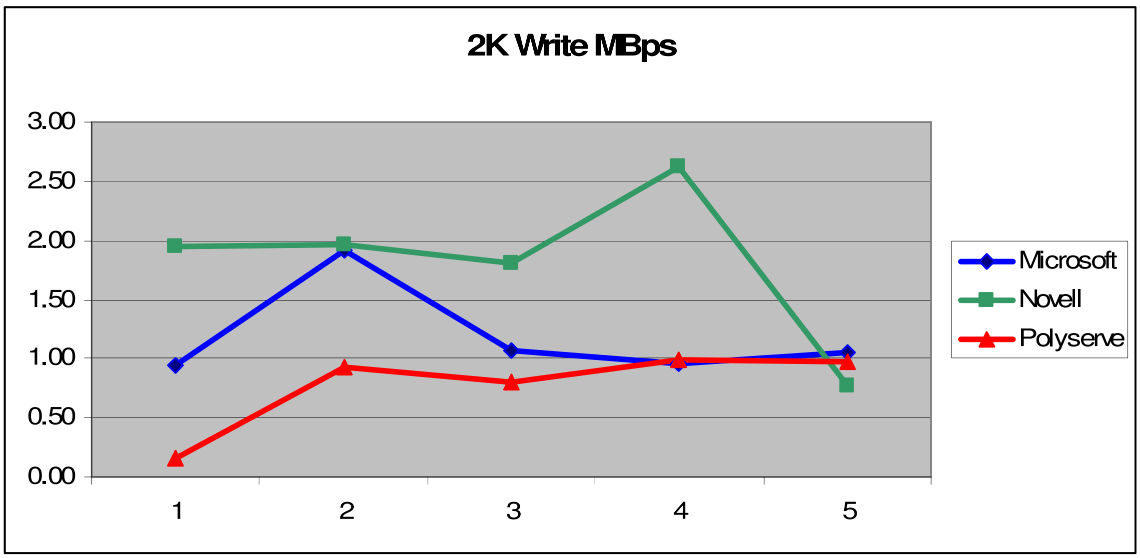
128K Write MBps	**1**	**2**	**3**	**4**	**5**
Microsoft	86.92	96.40	83.11	95.70	106.47
Novell	42.22	61.70	64.07	63.72	69.35
Polyserve	23.80	21.10	19.98	19.23	16.76

256K Read MBps	**1**	**2**	**3**	**4**	**5**
Microsoft	171.68	163.87	162.97	162.62	178.36
Novell	98.39	122.37	130.10	112.74	125.42
Polyserve	97.65	136.63	150.00	133.92	128.55

256K Write MBps	**1**	**2**	**3**	**4**	**5**
Microsoft	119.80	126.35	123.09	140.66	152.05
Novell	79.28	93.41	92.17	92.41	117.10
Polyserve	45.89	40.22	36.96	37.15	37.36

2K Read/Write



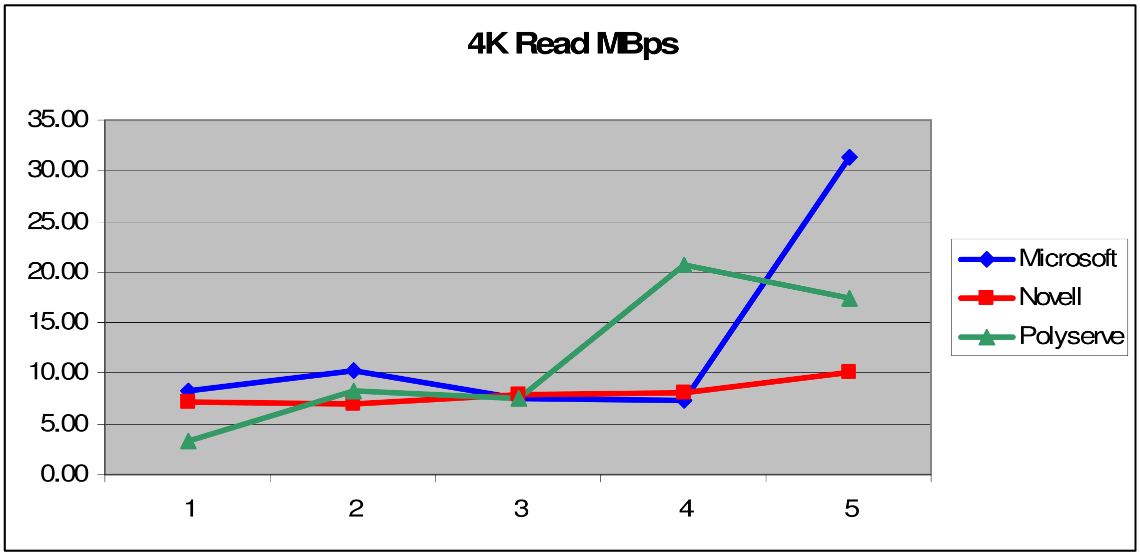


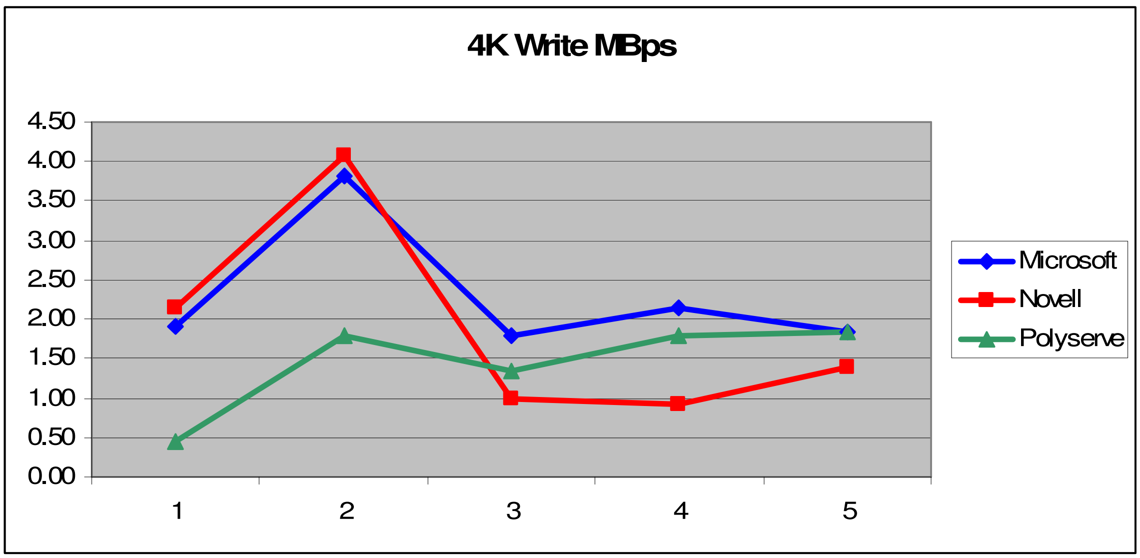
The first item of note in these tests is the difference between read and write performance. The Compellent is tuned for read performance, which generally is about ¾ of all I/O. The 2K blocks Read/Write ratio is 5.63:1.

Read performance changes relatively little with write caching, and dips with "Enable Writes to Read Cache" enabled.

Microsoft has the best overall performance for reads. Novell has the best overall performance for writes.

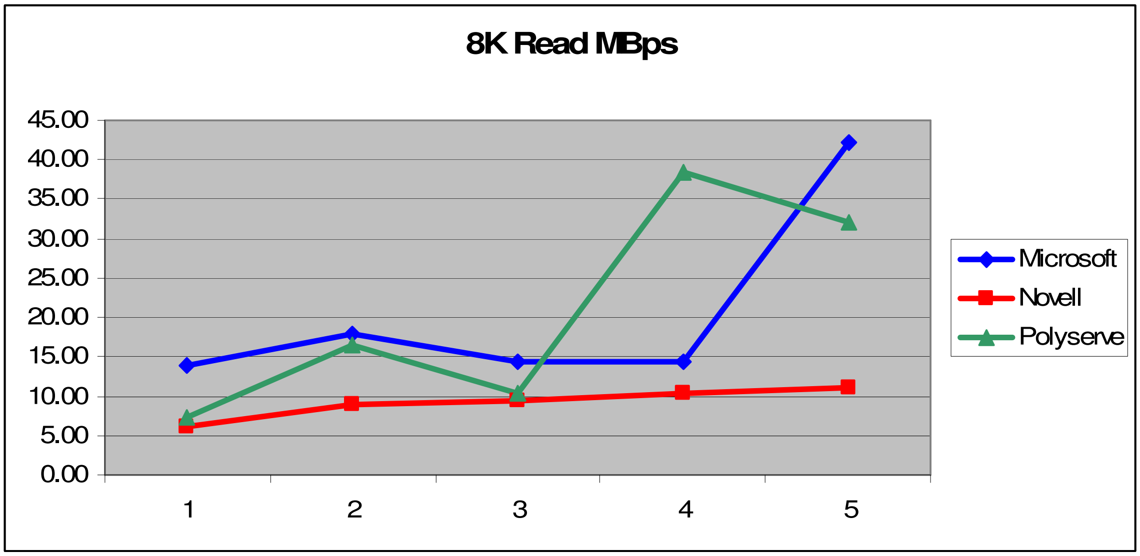
4K Read/Write

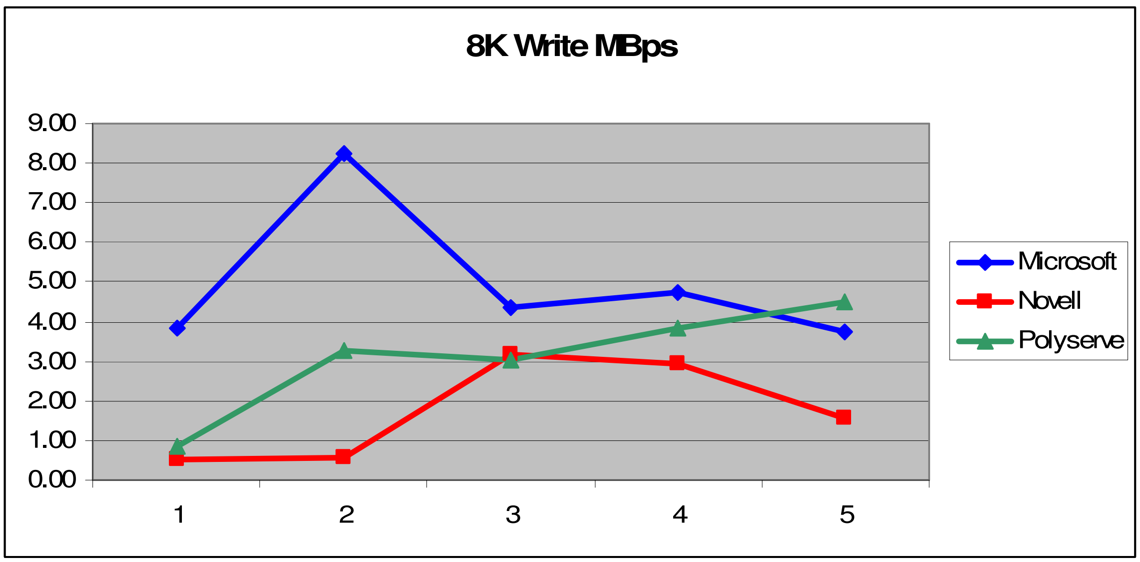




The Compellent 4K Read-Write ratio is 5.74:1. Overall, Microsoft offers the best performance on reads and writes. Novell's performance is slightly better without caching or with only write caching enabled.

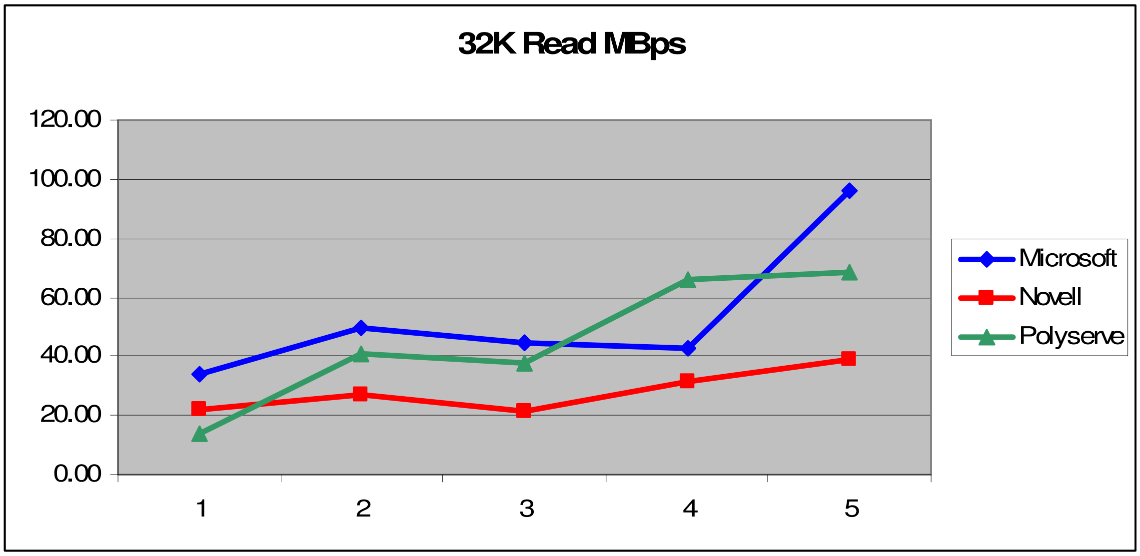
8K Read/Write

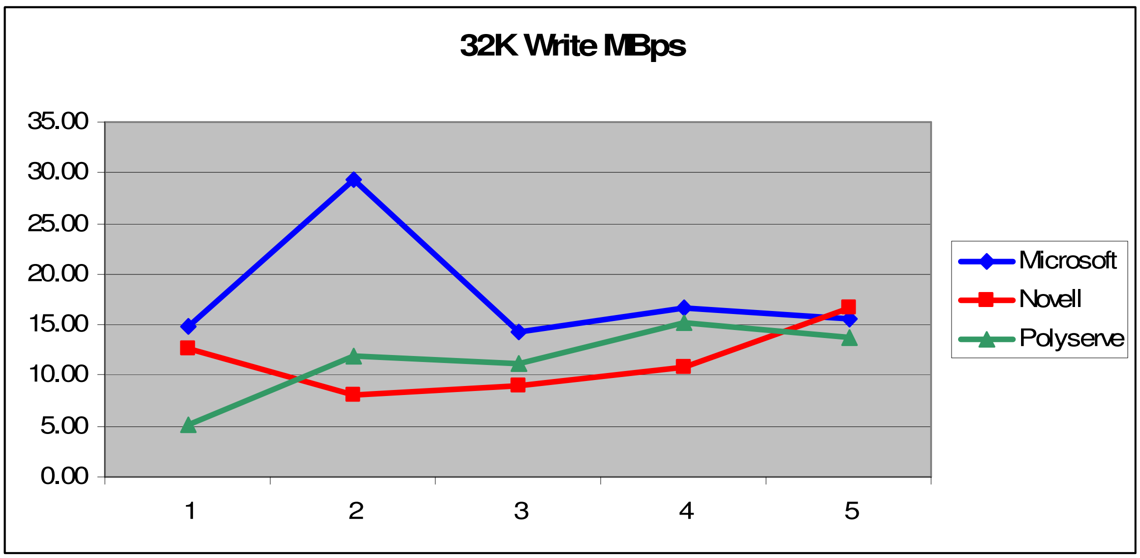




The Compellent 8K Read-Write ratio is 5.14:1. Microsoft has the overall performance. The caching changes had the biggest improvement with the Microsoft cluster.

32K Read/Write

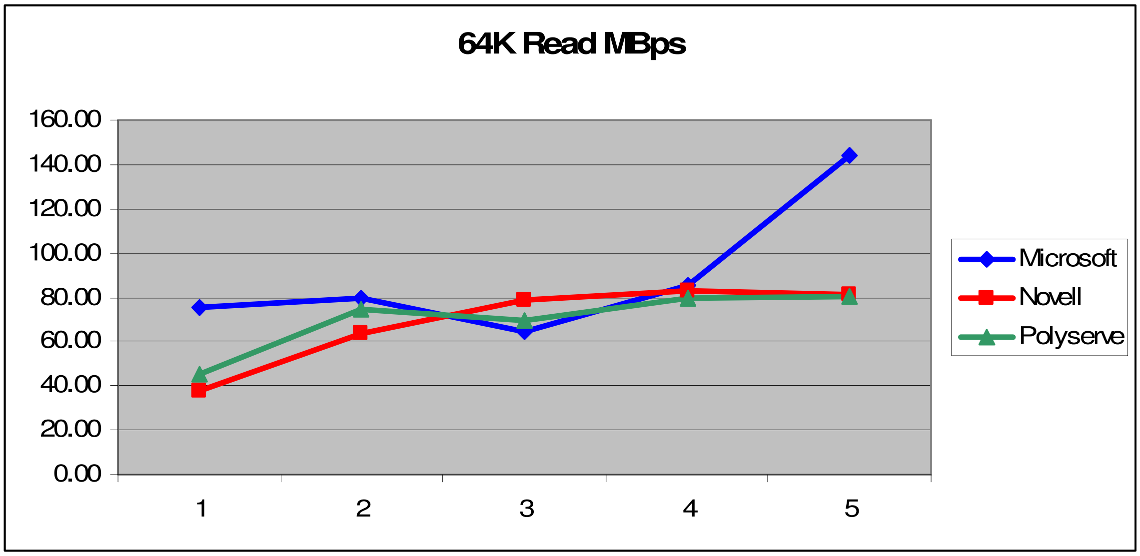


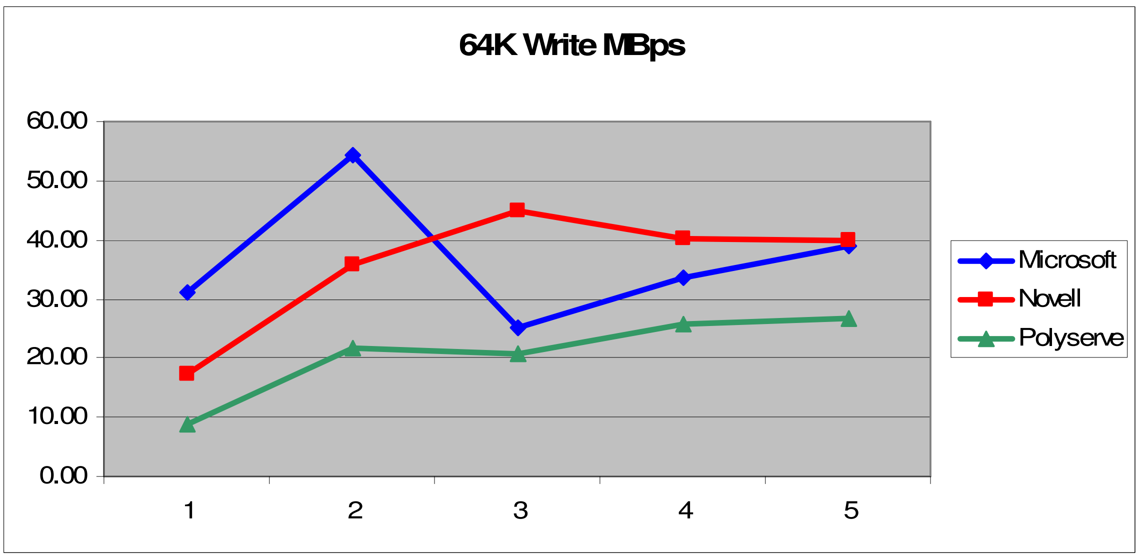


For the Compellent, 32K appears to be the breaking point for Read/Write tuning. Lower block I/O hovers around 5:1. Higher block I/O is closer to 2:1. In these tests, the 32K block Read/Write ratio is 3.1:1.

Microsoft has the best overall performance at 32K.

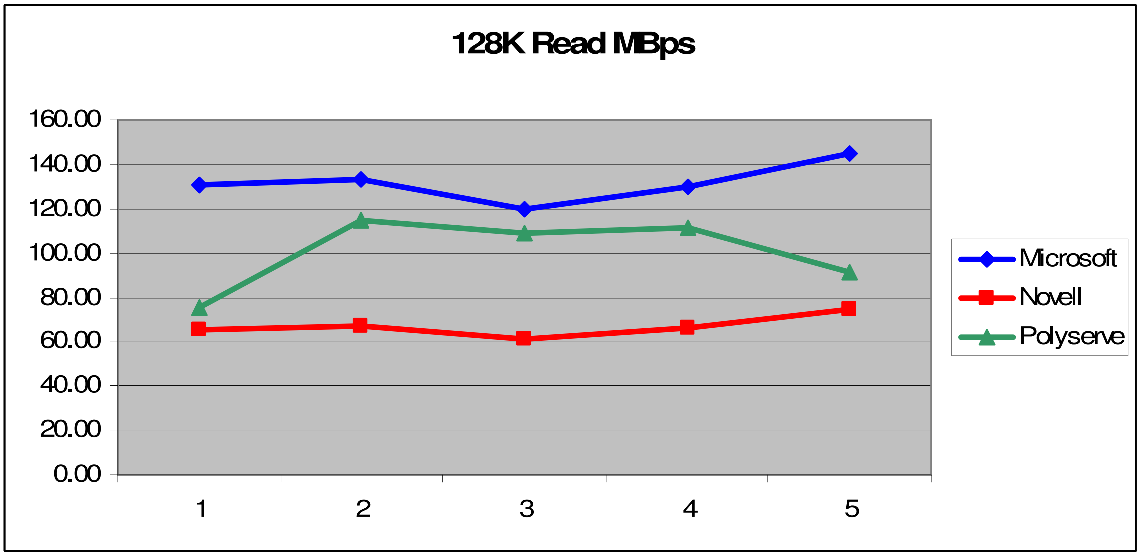
64K Read/Write

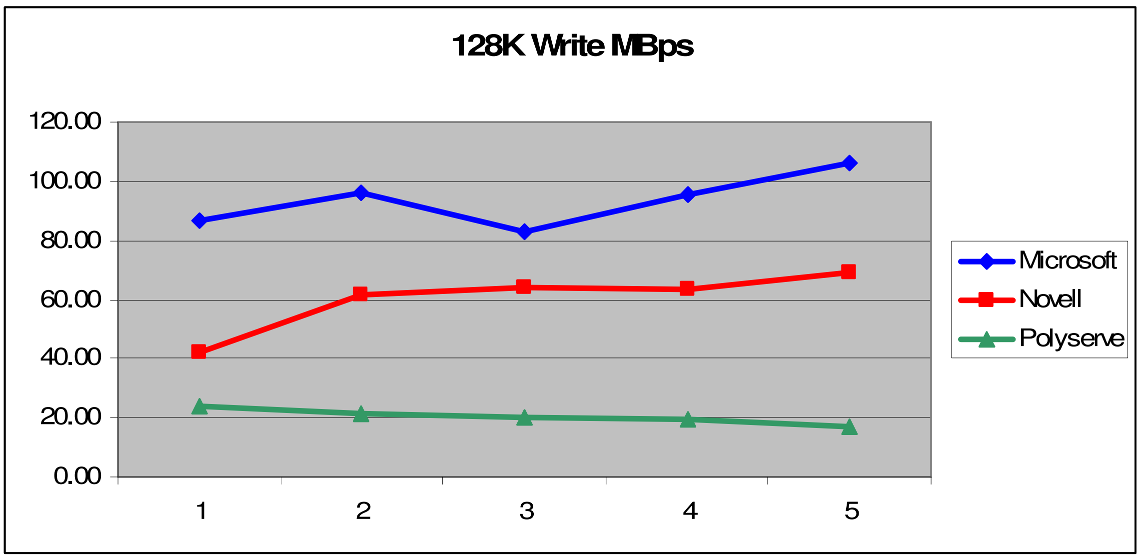




The Compellent 64K Read/Write ratio is 2.46:1. Microsoft has the best overall performance. In certain tests, Novell outperformed the other clustering technologies on writes.

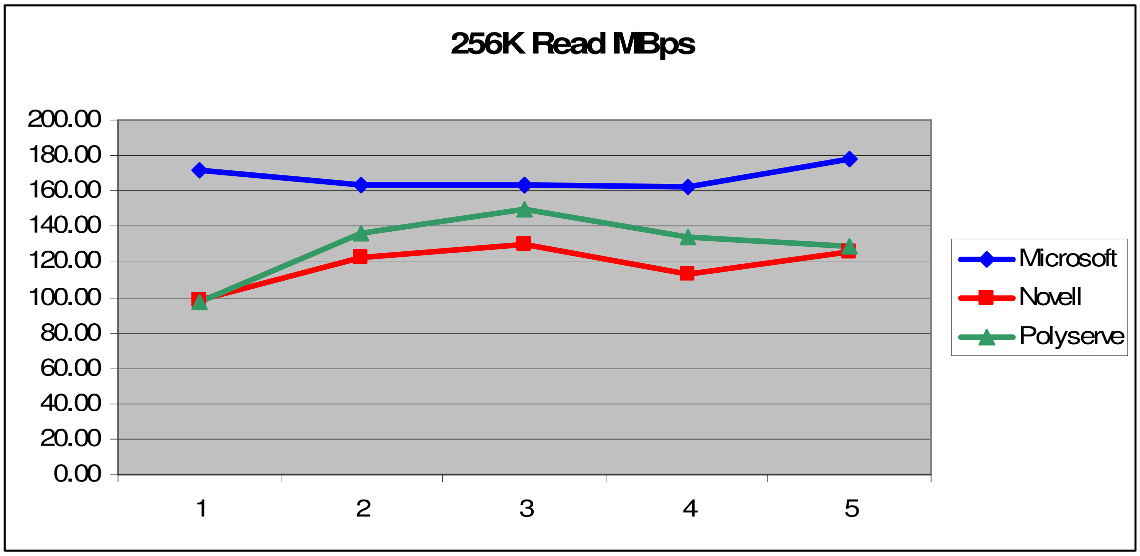
128K Read/Write

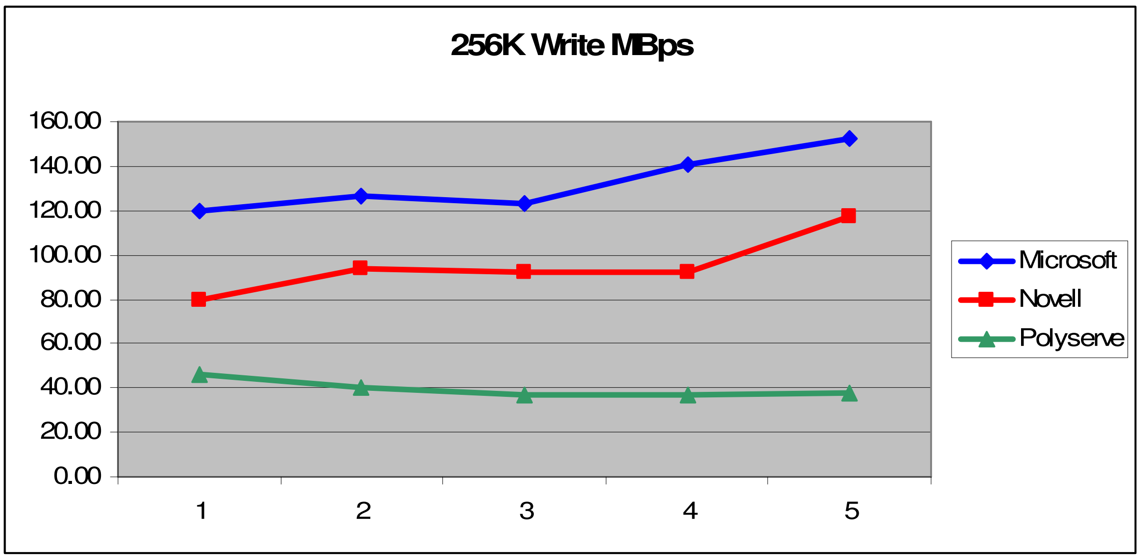




The Compellent 128K Read/Write ratio is 1.72:1. Microsoft has the best overall performance. Polyserve clearly has better read performance at the higher block I/O, while Novell has higher write performance.

256K Read/Write





The Compellent 256K Read/Write ratio is 1.56:1. Microsoft, again, has the best overall performance. Novell and Polyserve have similar read performance. Novell betters Polyserve in large I/O write speed, however.

Local I/O Test 2

This data represents five tests, varying the NumberOfRequests and outstanding I/O from 16-255.

2K Read MBps	**16**	**32**	**64**	**128**	**255**
Microsoft	4.56	5.19	5.24	3.89	21.13
Polyserve	9.14	9.58	11.38	12.21	12.80

2K Write MBps	**16**	**32**	**64**	**128**	**255**
Microsoft	0.95	1.92	1.07	0.95	1.04
Polyserve	0.97	0.91	0.77	1.03	1.02

4K Read MBps	**16**	**32**	**64**	**128**	**255**
Microsoft	8.30	10.20	7.51	7.26	31.28
Polyserve	17.44	18.93	22.84	23.15	24.91

4K Write MBps	**16**	**32**	**64**	**128**	**255**
Microsoft	1.90	3.83	1.80	2.14	1.83
Polyserve	1.85	1.77	2.32	2.00	2.27

8K Read MBps	**16**	**32**	**64**	**128**	**255**
Microsoft	13.87	17.92	14.35	14.32	42.22
Polyserve	31.97	44.01	41.57	36.08	41.33

8K Write MBps	**16**	**32**	**64**	**128**	**255**
Microsoft	3.83	8.23	4.37	4.74	3.76
Polyserve	4.49	4.86	4.34	4.45	4.96

32K Read MBps	**16**	**32**	**64**	**128**	**255**
Microsoft	34.15	49.44	44.33	42.63	96.12
Polyserve	68.79	89.66	62.33	81.37	80.79

32K Write MBps	**16**	**32**	**64**	**128**	**255**
Microsoft	14.87	29.31	14.29	16.72	15.53
Polyserve	13.74	20.67	16.21	18.21	17.10

64K Read MBps	**16**	**32**	**64**	**128**	**255**
Microsoft	75.72	79.95	64.80	85.13	143.83
Polyserve	80.11	111.93	77.25	107.90	118.01

64K Write MBps	**16**	**32**	**64**	**128**	**255**
Microsoft	31.05	54.46	25.00	33.57	39.07
Polyserve	26.76	35.88	25.87	31.73	30.41

128K Read MBps	**16**	**32**	**64**	**128**	**255**
Microsoft	130.36	133.19	119.83	129.82	144.92
Polyserve	91.61	150.34	152.14	152.25	152.42

128K Write MBps	16	32	64	128	255
Microsoft	86.92	96.40	83.11	95.70	106.47
Polyserve	16.76	32.83	34.59	37.01	39.57

256K Read MBps	16	32	64	128	255
Microsoft	171.68	163.87	162.97	162.62	178.36
Polyserve	128.55	162.02	167.34	141.51	154.57

256K Write MBps	16	32	64	128	255
Microsoft	119.80	126.35	123.09	140.66	152.05
Polyserve	37.36	38.30	40.19	70.00	87.37

Rate of Change

2K Read MBps	16	32	64	128	255	Total
Average	6.85	7.38	8.31	8.05	16.97	9.51
% Change		8%	13%	-3%	111%	148%

2K Write MBps	16	32	64	128	255	Total
Average	0.96	1.41	0.92	0.99	1.03	1.06
% Change		48%	-35%	8%	4%	8%

4K Read MBps	16	32	64	128	255	Total
Average	12.87	14.57	15.18	15.21	28.09	17.18
% Change		13%	4%	0%	85%	118%

4K Write MBps	16	32	64	128	255	Total
Average	1.87	2.80	2.06	2.07	2.05	2.17
% Change		49%	-26%	0%	-1%	9%

8K Read MBps	16	32	64	128	255	Total
Average	22.92	30.96	27.96	25.20	41.77	29.76
% Change		35%	-10%	-10%	66%	82%

8K Write MBps	16	32	64	128	255	Total
Average	4.16	6.54	4.35	4.60	4.36	4.80
% Change		57%	-33%	6%	-5%	5%

32K Read MBps	16	32	64	128	255	Total
Average	51.47	69.55	53.33	62.00	88.46	64.96
% Change		35%	-23%	16%	43%	72%

32K Write MBps	16	32	64	128	255	Total
Average	14.30	24.99	15.25	17.46	16.32	17.66
% Change		75%	-39%	15%	-7%	14%

64K Read MBps	16	32	64	128	255	Total
Average	77.92	95.94	71.02	96.51	130.92	94.46
% Change		23%	-26%	36%	36%	68%

64K Write MBps	16	32	64	128	255	Total
Average	28.90	45.17	25.43	32.65	34.74	33.38
% Change		56%	-44%	28%	6%	20%

128K Read MBps	16	32	64	128	255	Total
Average	110.99	141.77	135.99	141.04	148.67	135.69
% Change		28%	-4%	4%	5%	34%

128K Write MBps	16	32	64	128	255	Total
Average	51.84	64.62	58.85	66.36	73.02	62.94
% Change		25%	-9%	13%	10%	41%

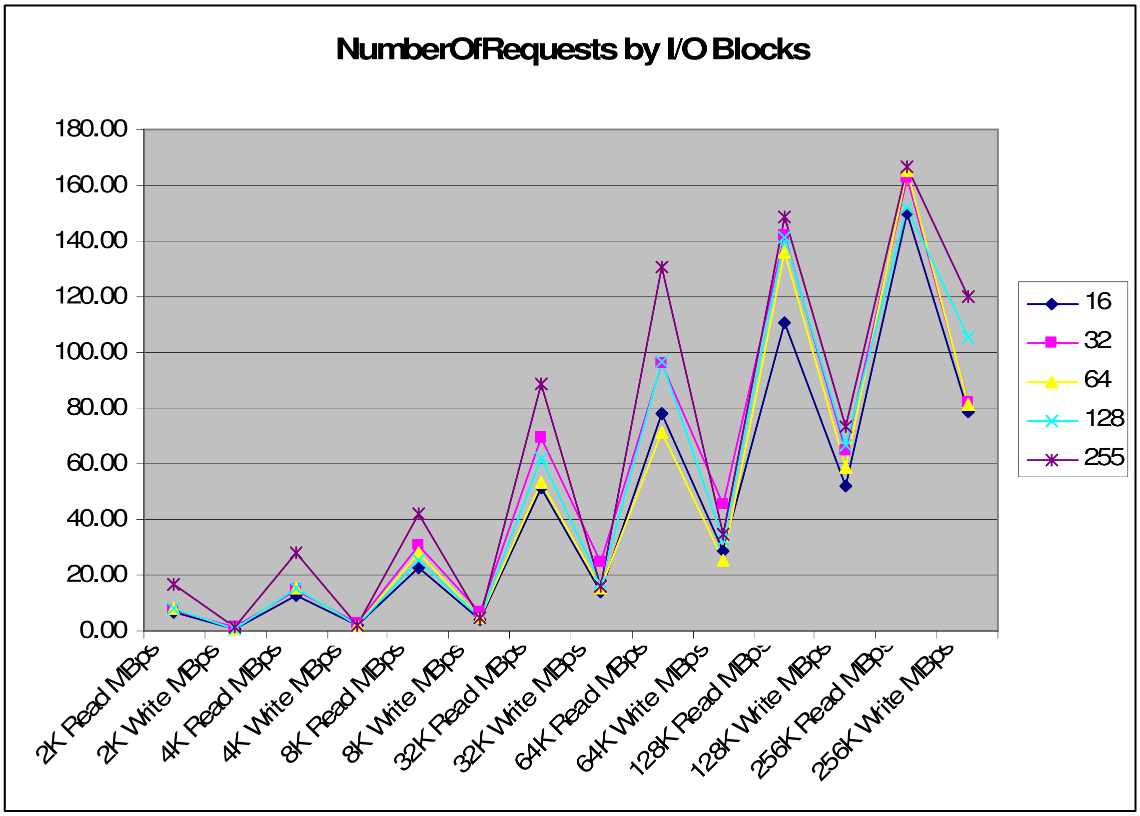
256K Read MBps	16	32	64	128	255	Total
Average	150.12	162.94	165.16	152.07	166.47	159.35

% Change		9%	1%	-8%	9%	11%

256K Write MBps	**16**	**32**	**64**	**128**	**255**	**Total**
Average	78.58	82.32	81.64	105.33	119.71	93.52
% Change		5%	-1%	29%	14%	52%

The largest jump in performance was from 16-32 requests. From 32-64 often resulted in a drop in overall performance.

Read I/O benefited from the increased queue far more than write I/O. Overall, reads increased over 500%, while writes increased only 150%.



Network I/O Test

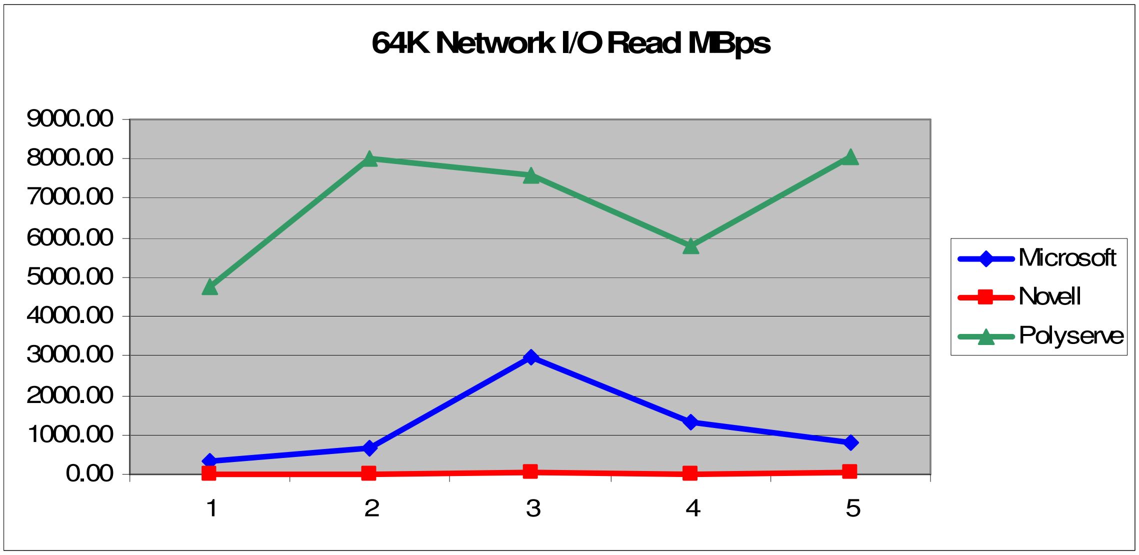
64K Read MBps	1	2	3	4	5
Microsoft	350.55	667.82	2964.92	1333.53	793.12
Novell	9.67	15.46	32.81	13.96	33.50
Polyserve	4755.25	8025.39	7587.70	5784.09	8067.50

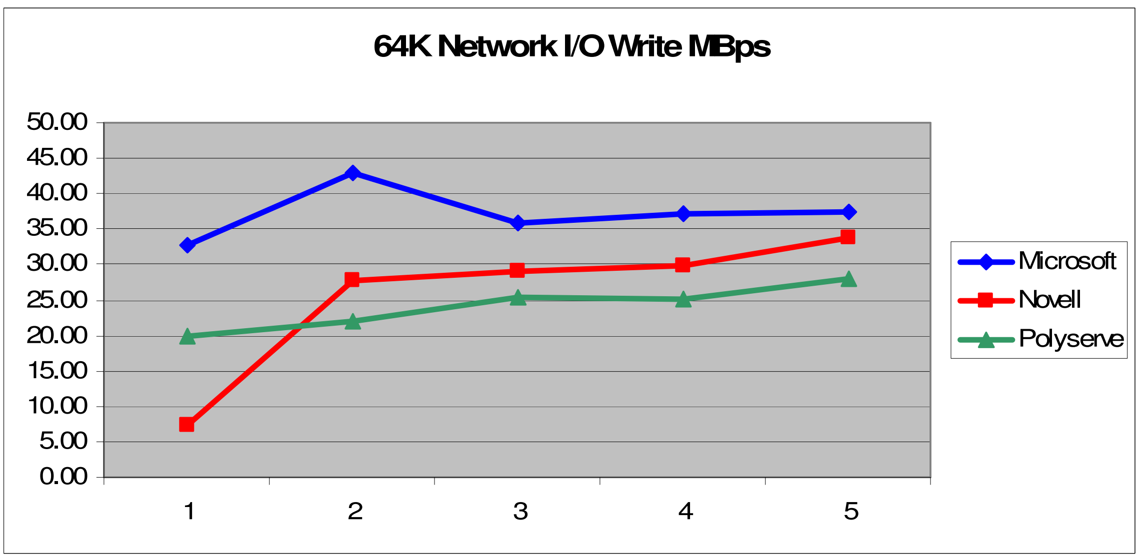
64K Write MBps	1	2	3	4	5
Microsoft	32.73	42.99	35.86	37.24	37.38
Novell	7.38	27.82	29.07	29.96	33.77
Polyserve	19.88	21.97	25.34	25.19	28.00

Estimated Actual Disk Performance

64K Read MBps	1	2	3	4	5
Microsoft	81.00	106.00	89.00	92.00	92.00
Novell	19.00	69.00	72.00	74.00	84.00
Polyserve	49.00	55.00	63.00	62.00	69.00

64K Read/Write





The measured I/O read performance is significantly faster than any other observed metric. This appears to be due to the caching inherent with CIFS. The Iozone documentation states that the "performance of a system under this type of activity can be impacted by several factors such as: Size of operating system's cache, number of disks, seek latencies, and others."

Microsoft and Polyserve use CIFS for file transfers and Novell uses NCP. This difference may explain why Polyserve is at 4700-806 MBps, Microsoft is at 350-2900 MBps, and Novell has rates of 9-34 MBps.

Of note, Novell could not be tested using all 24 workstations. Such a test appeared to saturate the Netware system, making it unresponsive to requests. We scaled down to between 6-12 workstations. Even at this rate, many test samples were lost because the Netware system refused connections.

We can estimate the actual I/O to disk rate by using the observed Compellent 64K Read/Write ratio. This estimation comes close to the results seen in the Local I/O tests (+/- 6%), and to the throughput observed on the Compellent storage system at the time of the test.

The Network I/O writes also are close to the values observed for local I/O (+/- 7%).

Section 4: Summary

Where is the bottleneck?

If the client Lan connection is the bottleneck, then the performance would be limited by its maximum bandwidth. A 100 Mbps Full connection, less the overhead of Ethernet, IP, TCP, and CIFS, has a potential write bandwidth of 11.56 MBps. We are able to observe close to this –10-11 MBps throughput – when testing with 1-3 clients. The performance does not scale linearly when testing exceeds 3 clients or 30 MBps.

If the server Lan connection is the bottleneck, again, the performance would limited by its bandwidth. The 1 Gbps Full connection conceivably offers 10x the bandwidth of 100 Mbps Full, or 115.58 MBps. The observed throughput is significantly less than this.

Given that the observed network I/O is similar to the local I/O, that is the Lan tests mirror the San tests, it is apparent that the bottleneck lies with the storage system. The testing supports the hypothesis that the performance bottleneck is with the disk I/O.

Which clustering technology offers the best performance in this environment?

The Compellent storage system is highly utilized, particularly during the times that these tests were performed. This adds latency to each I/O request. This explains the boost in performance for higher block I/O – less latency per request means significantly higher throughput.

Microsoft Windows Clustering consistently offered the best overall performance.

PolyServe Matrix Server offered adequate performance. However, it would be difficult to justify the additional cost on performance alone. It was curious to see that the NTFS outperformed PSFS. One thought is because extra I/O necessary to maintain the journaling information. Subsequent tests run on a dedicated storage system (outside of the scope of this paper) showed no significant difference between NTFS and PSFS performance.

Overall, Novell Netware showed the lowest performance. That the network I/O tests had to be scaled back from 24 to 10 workstations is particularly troubling. Again, this may be explained by the difference in how the file systems handle I/O. NSS (Novell Storage Services) requires greater San IOps to process incoming Lan I/O. Another item of note is that Netware's primary performance boost comes from its caching technology, which these tests do not adequately reflect.

The recommendation is to go with Microsoft clustering at this time. When the Compellent storage system can consistently deliver 100 MBps or greater, the bottleneck will move to the Lan. PolyServe should be revisited at this point as its active-active configuration may offer better performance in this scenario.