

SMB3 Application Workload Performance

Tom Talpey
Architect, Windows File Server

Full credit to: Dan Lovinger
Windows File Server Performance

Microsoft

Outline

- SMB3 Application workloads
- Methodology
- Results and Discussion
- Work-in-progress

Goals

- Demonstrate SMB3 is valid Best Choice for application workloads
- Evaluate potential of new server hardware with SMB3
- Evaluate performance of RDMA-capable fabric(s)
- Measure the comparison to local storage
- Demonstrate that it is reasonable to consider remotely deployed storage for highly scaled server environments.
- Chart a future performance course, and metrics to use

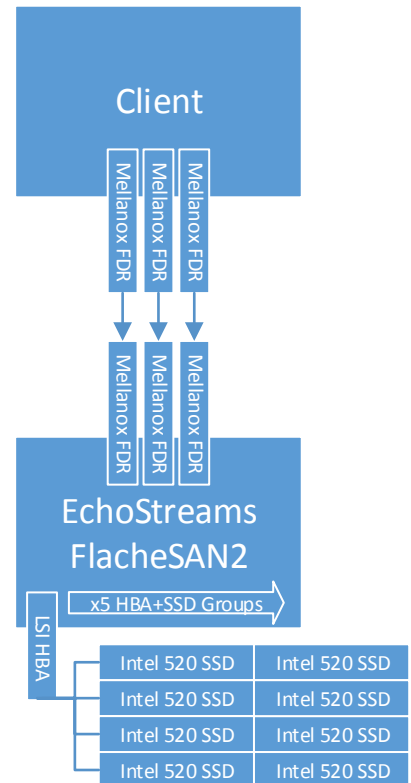
Key SMB3 Application Workloads

- Hyper-V (virtualization)
- SQL

- 8K Random
 - VHDs and database tables
 - Pure read, plus read/write mix
- 512K Sequential
 - Backup, disk migration, decision support/data mining
 - Pure read
 - Can be >512K, but performance and requirements largely the same
 - Also 64K

EchoStreams FlacheSAN2

- Appliance combining SAS HBAs, enterprise SSDs and high speed networking, Windows Server 2012 and Storage Spaces.
 - Networking: three Mellanox ConnectX-3 FDR InfiniBand HCAs
 - Storage
 - 5x LSI PCIe Gen 3.0 SAS HBA (6 possible)
 - 8x Intel 520 SSDs per controller
 - Total: five groups of eight for 40 total SSDs (48 possible)
 - 5x mirrored 4-column 2-copy Space, exposed as SMB3 shares
 - CPU: 2x Intel Xeon E5-2650 (8c16t 2.00Ghz)
 - Latest version available with 2.40GHz CPUs
 - DRAM: 32GB
- Client generic white box
 - Networking: three Mellanox ConnectX-3 FDR InfiniBand HCAs
 - CPU: 2x Intel Xeon E5-2680 (8c16t 2.70Ghz)
 - DRAM: 128GB



Methodology

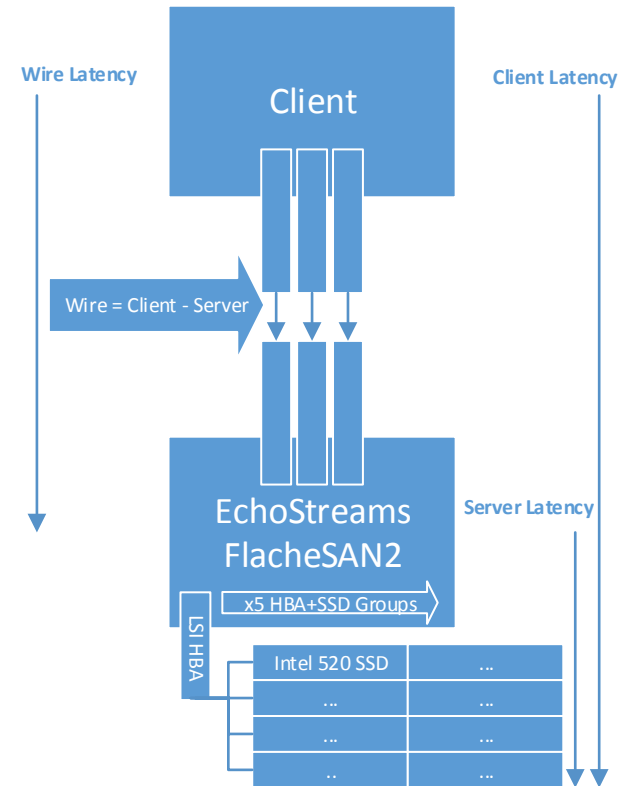
- Client workload generator: Microsoft sqlio.exe
 - Affinitized to run on specific CPU cores
 - Two instances, one per socket
- Server virtual drives
 - Each share exposes two 100GB files
 - Client instances split load per-socket
- Goal to emulate typical NUMA-aware modern application
 - E.g. Windows Hyper-V, guests running with affinity to specific socket(s) and core(s), accessing per-VM VHDs
- Units:
 - KB/MB/GB = **decimal**: $10^3/10^6/10^9$
 - KiB/MiB/GiB = **IEC60027-2**: $2^{10}/2^{20}/2^{30}$

Metric: Overhead

- Cycles/Byte
 - Standard measure of CPU bandwidth efficiency
 - $$c/B = \frac{\%Privileged\ CPU\ Utilization \times Core\ Clock\ Frequency \times \#Cores}{Bandwidth\ in\ Bytes}$$
- Privileged CPU utilization from Windows Performance counters
 - Discounts any unrelated activity, and from load generator itself
- Core clock is not constant - must configure system under test to minimize processor frequency variation:
 - Hyperthreading disabled
 - TurboBoost and SpeedStep disabled
 - Virtualization disabled
 - BIOS deep C-states disabled
 - Windows power plan to Max Performance
- Re-enabling can improve performance, i.e. results are conservative.

Metric: Latency

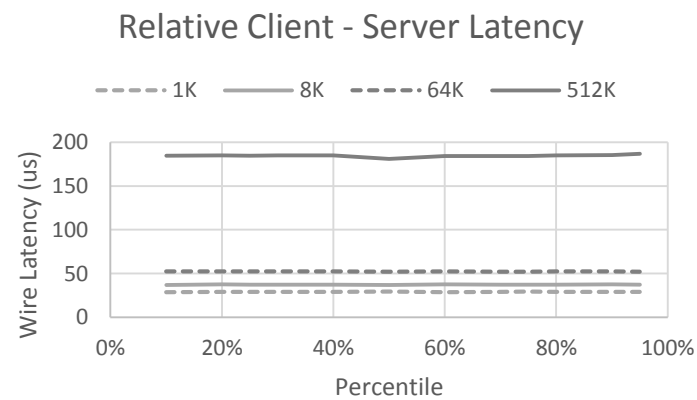
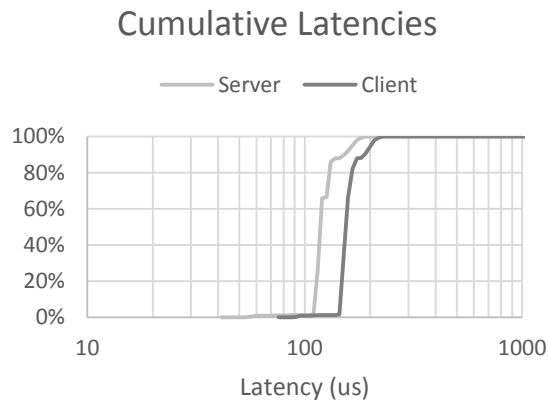
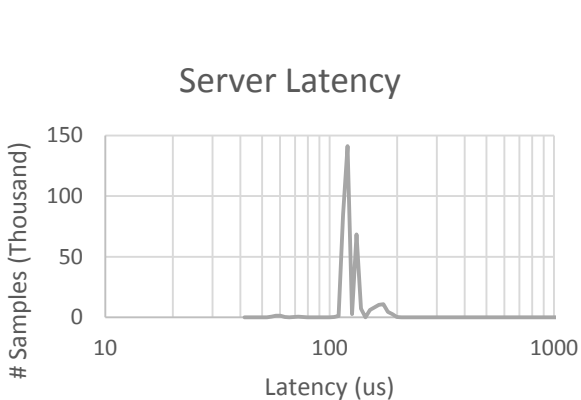
- Two client-visible components of latency:
 - Wire
 - Bit transmission time
 - Includes request queuing on/off adapter
 - Visible in Windows perfmon “stalls”
 - Server
 - Filesystem (NTFS) processing time
 - Storage processing time
- Measured as 90th percentile
 - Captured with Windows Performance Analyzer
 - Individual I/O latencies
 - 1M samples or 1 minute, with warm-up
- Unexpected latency increase can indicate bottleneck being reached
 - E.g. CPU saturation or other overhead



Result 1: Single I/O Latency

- Single random I/O to single share
- Used to establish base latency expected of systems
- Consistent, good performance, exposing wire and SSD latencies

Latency (us)	90th Percentile Read			90th Percentile Write		
Size (KiB)	Client	Server	Wire	Client	Server	Wire
1	204	176	29	153	119	34
8	197	159	38	113	65	49
64	419	366	52	366	303	63
512	1297	1112	185	1355	1143	212

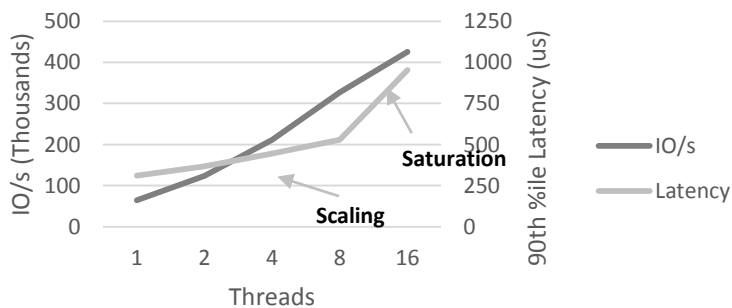


Result 2: Small I/O Scaling - Read

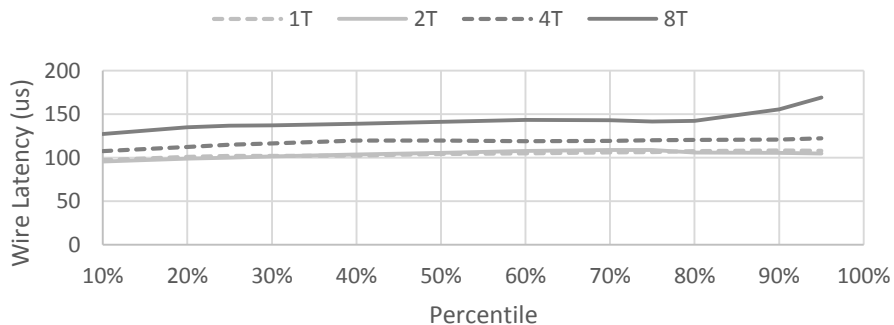
- Client CPU comfortable
- Server CPU saturates at high thread count
 - Note relatively low server CPU clock (2.00 GHz)

	1KiB					8KiB				
Threads	IOPs	90 th (us)	c/B	%CPU	%CPU Srv	IOPs	90 th (us)	c/B	%CPU	%CPU Srv
1 (20 I/O)	76650	265	43.3	7.9	10.8	64500	310	7.9	9.7	9.8
2 (40 I/O)	144050	320	43.3	14.8	21.7	123600	365	7.0	16.4	20.3
4 (80 I/O)	244250	390	41.4	24.0	48.4	211500	445	6.5	26.3	46.6
8 (160 IO)	360950	560	41.7	35.7	84.5	327050	530	6.4	40.0	82.5
16 (320 IO)	438400	1040	44.9	46.6	99.9	425900	955	7.2	58.2	100.0

Scaled 8KiB Read at 20 IO/Thread



Scaled 8KiB Read Wire Latency at 20 IO/Thread

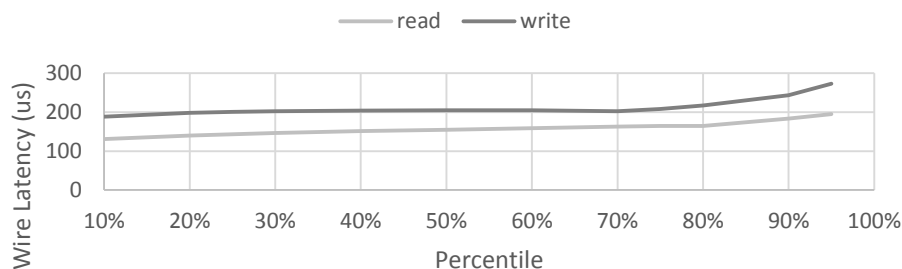


Result 3: Small I/O Scaling – 60R/40W Mixed

- Similar to read
- As expected, since load is not bandwidth-limited
 - Scaling may increase on bi-directional links, if available

Threads	1KiB					8KiB				
	IOPs	90th R(us)	90th W(us)	c/B	%CPU	IOPs	90th R(us)	90th W(us)	c/B	%CPU
1 (20 I/O)	69800	300	350	44.3	7.3	70700	310	270	7.1	9.5
2 (40 I/O)	125900	355	410	45.3	13.5	124950	370	340	7.0	16.6
4 (80 I/O)	206450	435	495	43.4	21.3	210850	450	410	6.9	27.4
8 (160 I/O)	319150	545	635	39.6	30.0	328150	575	510	6.8	42.5
16 (320 I/O)	424850	960	1140	47.1	47.5	375900	1235	1330	7.4	52.7

Mixed 8KiB Wire Latency at 8T 20 IO/Thread

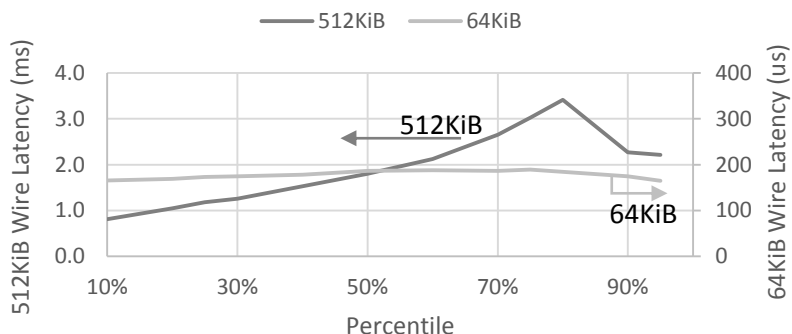


Result 4: Large I/O (Read)

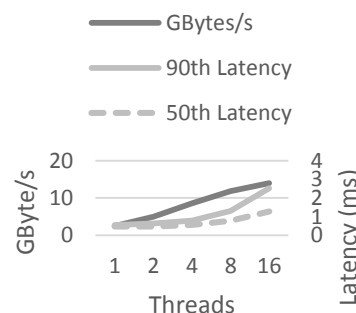
- Full bandwidth (16+ GBps!) achievable, at very low CPU
- 512KB reaches limit of network at just under 16 threads
 - Multichannel round-robin leads to some latency variation near limit
 - CPU limit much better behaved, by comparison 😊

	64KiB				512KiB			
Threads	GBytes/s	90 th (us)	c/B	%CPU	GBytes/s	90 th (us)	c/B	%CPU
1 (20 I/O)	2.45	550	1.22	6.9	6.64	1630	0.31	4.7
2 (40 I/O)	4.95	630	1.06	12.2	11.34	2570	0.29	7.6
4 (80 I/O)	8.58	780	1.05	20.8	14.41	4970	0.29	9.8
8 (160 I/O)	11.84	1300	1.06	29.0	15.68	9930	0.30	10.9
16 (320 I/O)	13.99	2520	1.09	35.3	16.40	19900	0.31	11.6

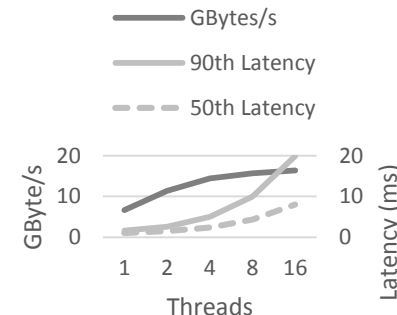
Large Read Wire Latency at 8T 20 IO/Thread



Scaling of 64KiB IO



Scaling of 512KiB IO



Conclusions (Windows Server 2012)

- Maximum bandwidth
 - **16.4GB/s** (~5.5GB/s/adapter)
 - 0.31 c/B overhead
 - For 512KiB I/Os
- High IOPS to real storage
 - **376,000** to FlacheSAN2
 - 6.4 c/B overhead
 - For 8KiB I/Os
- Near-constant latency profile

Work In Progress – Small I/O

8KiB Random Read	WS2012 IOPS*	WS “WIP” IOPS*	Δ IOPS	Δ c/B
1x54Gbps NIC	~330,000	~460,000	+36%	-17%
2x54Gbps NIC	~660,000	~860,000	+30%	-15%

* fictitious storage (/dev/zero)

- NOT FINAL and NOT GUARANTEED
- Intermediate results from local-only internal optimizations
 - Enhanced NUMA awareness
 - Improved request batching, locking, cacheline false sharing, etc
- Future improvements expected from
 - Further optimizations
 - Use of iWARP/InfiniBand remote invalidation
 - Watch for news on these at Storage Developers Conference!

References

- Windows Server 2012 EchoStreams FlacheSAN2 (paper)
 - <http://www.microsoft.com/en-us/download/details.aspx?id=38432>
- EchoStreams FlacheSAN2
 - <http://www.echostreams.com/flachesan2.html>
- SMB 3.0 Specification (MS-SMB2)
 - <http://msdn.microsoft.com/en-us/library/cc246482.aspx>
- SMB Direct Specification (MS-SMBD)
 - <http://msdn.microsoft.com/en-us/library/hh536346.aspx>
- Windows Performance Analyzer
 - <http://go.microsoft.com/fwlink/?LinkId=214551>
- Contact
 - ttalpey -at- microsoft.com