

# ZIL Accelerator: DRAM or Flash?

***DDRdrive***<sup>™</sup>  
*The drive for speed.*<sup>™</sup>

Christopher George  
Founder/CTO  
[www.ddrdrive.com](http://www.ddrdrive.com)



OpenStorage Summit 2010  
October 26-27, 2010  
Palo Alto, CA USA

# Storage Terminology/Nomenclature:

- ZFS (Zettabyte File System)
- ZIL (ZFS Intent Log) Accelerator
  - a.k.a. SLOG (Separate LOG) or Dedicated Log
- SSD (Solid-State Drive)
  - SSD Types (Primary Media):
    - Flash (NAND) Based
    - DRAM (Dynamic Random Access Memory) Based
  - SSD Form Factors:
    - 3.5"/2.5" HDD (Hard Disk Drive) Compatible
    - PCI Express Plug-in Card
- IOPS (Input/Output operations Per Second)

# The Filesystem Reinvented.

## ZFS Hybrid Storage Pool:

A pool (or collection) of high capacity, low cost, and low RPM HDDs accelerated with integrated support of both read and write optimized SSDs.

The key is both storage devices (HDD/SSD) work together as one to provide the capacity and cost per bit benefits of an HDD with the performance and power benefits of an SSD.

## ZIL Accelerator:

One of the two optional accelerators built-in to ZFS. Expected to be write optimized as it only accelerates synchronous writes. Therefore, the intended device must have both extremely low latency and high *sustained* write IOPS.

End purpose is to accelerate applications bound by synchronous writes (e.g. NFS, iSCSI, CIFS).

Can be created from either type of SSD (DRAM or Flash). Which SSD type to choose?

## Questions to be answered:

- What is the ZIL (ZFS Intent Log)?
- Common characteristics of both ZIL Accelerator SSD types?
- Why does the ZIL Accelerator attachment interface matter?
- ZIL Accelerator access pattern random and/or sequential?
- The untold truth about Flash SSD write IOPS degradation?
- How does IO/partition alignment affect IOPS performance?
- How do ZIL Accelerator SSD types compare and contrast?

# Questions to be answered:

- **What is the ZIL (ZFS Intent Log)?**
- Common characteristics of both ZIL Accelerator SSD types?
- Why does the ZIL Accelerator attachment interface matter?
- ZIL Accelerator access pattern random and/or sequential?
- The untold truth about Flash SSD write IOPS degradation?
- How does IO/partition alignment affect IOPS performance?
- How do ZIL Accelerator SSD types compare and contrast?

# What is the ZFS Intent Log (ZIL)?

- Logs all file system related system calls as transactions in host memory. If synchronous semantics apply (`O_SYNC`, `fsync()`...), transactions are also placed on stable (non-volatile) storage, so in the event of a host failure each can be replayed on the next reboot.
- Satisfies POSIX requirements for synchronous write transactions.
- Default implementation uses the pool for stable “on-disk” format. Optionally, a ZIL Accelerator can be used to increase performance.
- One ZIL per dataset (e.g. file system, volume), with one or more datasets per pool. A ZIL Accelerator is a pool assigned resource and thus shared by **all** datasets (ZILs) contained in that pool.
- Transactions are committed to the pool as a group (txg) and involve reading the ZIL “in-memory” representation and NOT the “on-disk” format. After the txg commits, the relevant ZIL (on-disk or optionally a ZIL Accelerator) blocks are released.

# What is a synchronous write transaction?

- Synchronous writes are forced to stable (non-volatile) storage prior to being acknowledged. Commonly initiated by setting `O_SYNC`, `O_DSYNC`, or `O_RSYNC` flag parameters when the target file was opened or by calling `fsync()`.
- Guarantees, upon a host power or hardware failure all writes successfully acknowledged prior are safely stored and unaffected.
- Critical Assumption: All relevant storage devices (including HBA Controller) and device drivers must properly implement the SCSI `SYNCHRONIZE_CACHE` or ATA `FLUSH CACHE` command by flushing any/all volatile caches to stable (non-volatile) storage.
- **WARNING:** Some storage devices ignore the cache flush command and are unable to correctly implement synchronous write semantics.
- **WARNING:** Do NOT set the system-wide "`zfs_nocacheflush`" tunable unless every system storage device's volatile cache is power protected.

## Questions to be answered:

- What is the ZIL (ZFS Intent Log)?
- **Common characteristics of both ZIL Accelerator SSD types?**
- Why does the ZIL Accelerator attachment interface matter?
- ZIL Accelerator access pattern random and/or sequential?
- The untold truth about Flash SSD write IOPS degradation?
- How does IO/partition alignment affect IOPS performance?
- How do ZIL Accelerator SSD types compare and contrast?

## Common characteristics of both ZIL Accelerator SSD types?

- The ZIL Accelerator is added to a pool, thus **shared** by all datasets (file systems, volumes, clones) contained within this pool.
- Device data integrity is paramount to operational correctness. Unless the ZIL Accelerator is mirrored, no ZFS checksum fallback is available.
- Requires a low latency, high **sustained** write IOPS capable device.
- Write IOPS intensive, never read unless at reboot (replay) and import.
- **ZFS does NOT support TRIM**, an issue for Flash SSDs but not DRAM SSDs.
- BONUS: By relocating the default ZIL from the pool, it reduces both pool block fragmentation and pool IO congestion, increasing all IO performance.
- **WARNING**: Device must correctly and consistently implement the SCSI SYNCHRONIZE\_CACHE or ATA FLUSH CACHE command for cache flush support.
- **WARNING**: Operational correctness (cache flush support) requires power protection of ALL on-board volatile caches. Most obvious with memory components, but also beware of controller based on-chip volatile caches.

# Flash SSDs which do NOT power protect on-board volatile caches!



- Intel X25-E (Extreme)
- Intel X25-M (Mainstream)
- Intel X25-V (Value)



## Flash SSDs which do power protect on-board volatile caches:



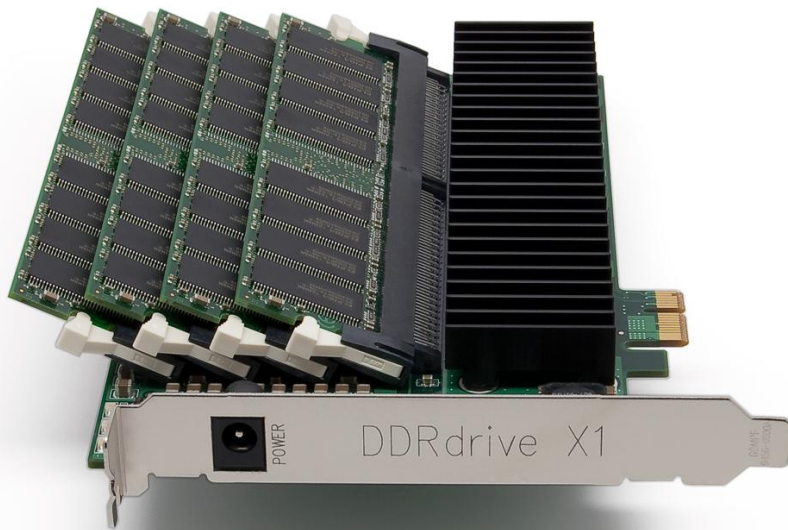
- OCZ Vertex 2 EX
  - Supercapacitor Backup
  - SLC (Single-Level Cell) Flash
  - SandForce 1500 Controller
  - 2.5" SATA II SSD



- OCZ Vertex 2 Pro
  - Supercapacitor Backup
  - MLC (Multi-Level Cell) Flash
  - SandForce 1500 Controller
  - 2.5" SATA II SSD

# DRAM SSD which power protects ALL on-board volatile memory.

## DDRdrive X1:



- Guarantees correct and consistent implementation of cache flushes. (SCSI SYNCHRONIZE\_CACHE command)
- Guarantees, in conjunction with an external UPS attached with the included ACDC adapter, all on-board volatile memory is power protected. During a host failure or power loss an automatic backup occurs transferring all DRAM contents to on-board SLC NAND. Automatically restores NAND to DRAM when host power recovers.
- Singularly designed to perform the unique function of a ZIL Accelerator.

## Questions to be answered:

- What is the ZIL (ZFS Intent Log)?
- Common characteristics of both ZIL Accelerator SSD types?
- **Why does the ZIL Accelerator attachment interface matter?**
- ZIL Accelerator access pattern random and/or sequential?
- The untold truth about Flash SSD write IOPS degradation?
- How does IO/partition alignment affect IOPS performance?
- How do ZIL Accelerator SSD types compare and contrast?

# Why does the ZIL Accelerator attachment interface matter?

BER = Bit Error Rate



No cable, no BER.



- Separate HBA/SSD
  - Storage Media
  - **SSD Controller**
  - **SATA/SAS Cable (BER)**
  - **HBA Controller**
  - PCIe Bus to Host (BER)
- Integrated HBA+SSD
  - Storage Media
  - **Unified Controller (No Cable)**
  - PCIe Bus to Host (BER)

# Why does the ZIL Accelerator attachment interface matter?

## Performance:

- Separate HBA/SSDs (higher latency)
  - Data path requires **two** separate hops to the Host.
  - Shared HBA controller is an IOPS bottleneck.
    - **How many SSDs can attach today and tomorrow?**
  - Storage protocol (SATA/SAS) overhead **cannot** be eliminated.
- Integrated HBA+Storage (lower latency)
  - Data path requires only **one** hop to the Host.
  - IOPS performance scales, on-board IOPS processor.
  - Storage protocol (SATA/SAS) overhead **is eliminated**.



# Why does the ZIL Accelerator attachment interface matter?

## Reliability:

- Separate HBA/SSDs (lower reliability)
  - Increased chance of controller failure (up to 2X).
  - Increased data path errors. **Every bus has a BER.**
  - Single point of failure (HBA), reduced redundancy.
- Integrated HBA+Storage (higher reliability)
  - Decreased chance of controller failure.
  - Decreased data path errors. **No bus, No BER.**
  - Nothing shared, increased redundancy.



# Questions to be answered:

- What is the ZIL (ZFS Intent Log)?
- Common characteristics of both ZIL Accelerator SSD types?
- Why does the ZIL Accelerator attachment interface matter?
- **ZIL Accelerator access pattern random and/or sequential?**
- The untold truth about Flash SSD write IOPS degradation?
- How does IO/partition alignment affect IOPS performance?
- Compare ZIL Accelerator SSD types by cost (IOPS/\$)?

## ZIL Accelerator Access Pattern?

The answer is a key variable in determining which of the SSD types is best suited as a ZIL Accelerator. As a Flash based SSD, unlike a DRAM SSD, has highly variable write IOPS performance depending on IO distribution (sequential, random, and mixed). For a Flash SSD, performance variability is especially pronounced if the workload is random or mixed. Contrast with a DRAM SSD, in which performance is absolutely consistent regardless of IO distribution.

# Is the ZIL Accelerator access pattern random and/or sequential?

iopattern.d - Single IOzone workload targeted at a single file system (ZIL):

DEVICE	%RAN	%SEQ	COUNT	MIN	MAX	AVG	KR	KW
sd5	6	94	152	4096	131072	34708	0	5152
sd5	0	100	506	4096	131072	7422	0	3668
sd5	0	100	830	4096	131072	7446	0	6036
sd5	2	98	272	4096	131072	21202	0	5632
sd5	1	99	483	4096	131072	8904	0	4200
sd5	0	100	606	4096	131072	8502	0	5032
sd5	1	99	511	4096	131072	12167	0	6072
sd5	1	99	440	4096	131072	10994	0	4724
sd5	0	100	601	4096	69632	8444	0	4956
sd5	1	99	583	4096	131072	12042	0	6856
sd5	1	99	436	4096	131072	10878	0	4632
sd5	2	98	148	4096	73728	18293	0	2644
sd5	0	100	928	4096	131072	7216	0	6540
sd5	6	94	152	4096	131072	34708	0	5152
sd5	2	98	544	4096	131072	9118	0	4844
sd5	0	100	928	4096	131072	7216	0	6540
sd5	2	98	414	4096	131072	16176	0	6540
sd5	1	99	267	4096	81920	11060	0	2884
sd5	0	100	943	4096	131072	7722	0	7112
sd5	5	95	152	4096	131072	34708	0	5152

# Is the ZIL Accelerator access pattern random and/or sequential?

seeksize.d - Single IOzone workload targeted at a single file system (ZIL):

ZIL Accelerator = sd5 (negative seek offsets)

value	----- Distribution -----	count
-32768		0
<b>-16384</b>		<b>35</b>
<b>-8192</b>		<b>234</b>
-4096		0
<b>-2048</b>		<b>70</b>
<b>-1024</b>		<b>35</b>
-512		0
<b>-256</b>		<b>3</b>
-128		0
-64		0
-32		0
-16		0
-8		0
-4		0
-2		0
-1		0
0	@@@	56701

# Is the ZIL Accelerator access pattern random and/or sequential?

seeksize.d - Single IOzone workload targeted at a single file system (ZIL):

ZIL Accelerator = sd5 (positive seek offsets)

value	----- Distribution -----	count
<b>0</b>	@@@	<b>56701</b>
1		0
2		0
4		0
<b>8</b>		<b>9</b>
<b>16</b>		<b>11</b>
<b>32</b>		<b>5</b>
<b>64</b>		<b>4</b>
<b>128</b>		<b>14</b>
256		0
512		0
<b>1024</b>		<b>35</b>
2048		0
4096		0
<b>8192</b>		<b>218</b>
<b>16384</b>		<b>1</b>
32768		0

# Is the ZIL Accelerator access pattern random and/or sequential?

iopattern.d - Five IOzone workloads each targeted at separate file systems (ZILs):

DEVICE	%RAN	%SEQ	COUNT	MIN	MAX	AVG	KR	KW
sd5	<b>71</b>	29	619	4096	131072	14862	0	8984
sd5	<b>63</b>	37	100	4096	131072	59064	0	5768
sd5	<b>27</b>	73	1706	4096	131072	5997	0	9992
sd5	<b>37</b>	63	717	4096	131072	12419	0	8696
sd5	<b>38</b>	62	488	4096	131072	19539	0	9312
sd5	<b>32</b>	68	962	4096	131072	10078	0	9468
sd5	<b>65</b>	35	820	4096	131072	10464	0	8380
sd5	<b>22</b>	78	946	4096	131072	12448	0	11500
sd5	<b>36</b>	64	1132	4096	131072	7927	0	8764
sd5	<b>55</b>	45	664	4096	131072	16414	0	10644
sd5	<b>22</b>	78	490	4096	131072	13642	0	6528
sd5	<b>30</b>	70	877	4096	131072	8322	0	7128
sd5	<b>42</b>	58	786	4096	131072	11886	0	9124
sd5	<b>21</b>	79	675	4096	131072	15316	0	10096
sd5	<b>33</b>	67	1628	4096	131072	7024	0	11168
sd5	<b>43</b>	57	458	4096	131072	24745	0	11068
sd5	<b>25</b>	75	459	4096	131072	14813	0	6640
sd5	<b>35</b>	65	1513	4096	131072	7607	0	11240
sd5	<b>52</b>	48	282	4096	131072	29441	0	8108
sd5	<b>28</b>	72	1677	4096	131072	7361	0	12056

# Is the ZIL Accelerator access pattern random and/or sequential?

seeksize.d - Five IOzone workloads each targeted at separate file systems (ZILs):

ZIL Accelerator = sd5 (negative seek offsets)

value	----- Distribution -----	count
-65536		12
-32768	@	9094
-16384	@	4162
-8192		2328
-4096		1210
-2048		824
-1024		695
-512		730
-256		2076
-128	@	3498
-64	@	3548
-32	@	6635
-16	@@	12743
-8		0
-4		0
-2		0
-1		0
0	@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@	158590

# Is the ZIL Accelerator access pattern random and/or sequential?

seeksize.d - Five IOzone workloads each targeted at separate file systems (ZILs):

ZIL Accelerator = sd5 (positive seek offsets)

value	----- Distribution -----	count
0	@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@	158590
1		0
2		0
4		0
8	@@@	18452
16	@@	10456
32	@	5298
64	@	3767
128	@	3641
256		2121
512		952
1024		743
2048		852
4096		1178
8192		2258
16384	@	4132
32768	@	9035
65536		0

# Is the ZIL Accelerator access pattern random and/or sequential?

## ZIL Accelerator Access Pattern:

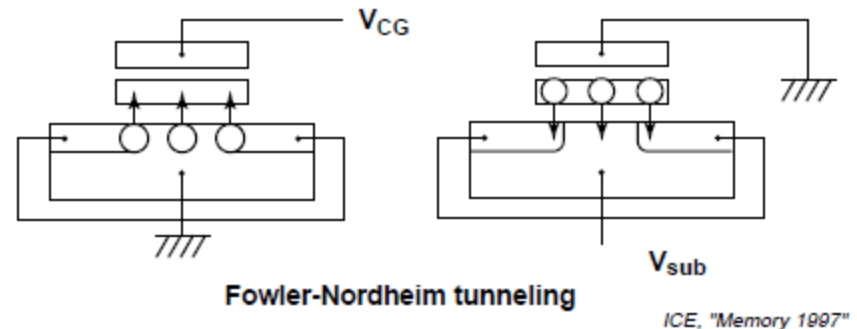
A predominately sequential write pattern is found for a pool with only a single file system. But as additional file systems are added to the pool, the resultant (or aggregate) write pattern trends to random access. Almost 50% random with a pool containing just 5 filesystems. This makes intuitive sense knowing each filesystem has a ZIL and **all** share the pool assigned ZIL Accelerator.

## Questions to be answered:

- What is the ZIL (ZFS Intent Log)?
- Common characteristics of both ZIL Accelerator SSD types?
- Why does the ZIL Accelerator attachment interface matter?
- ZIL Accelerator access pattern random and/or sequential?
- **The untold truth about Flash SSD write IOPS degradation?**
- How does IO/partition alignment affect IOPS performance?
- How do ZIL Accelerator SSD types compare and contrast?

# Thought Experiment: What is the underlying physics of Flash?

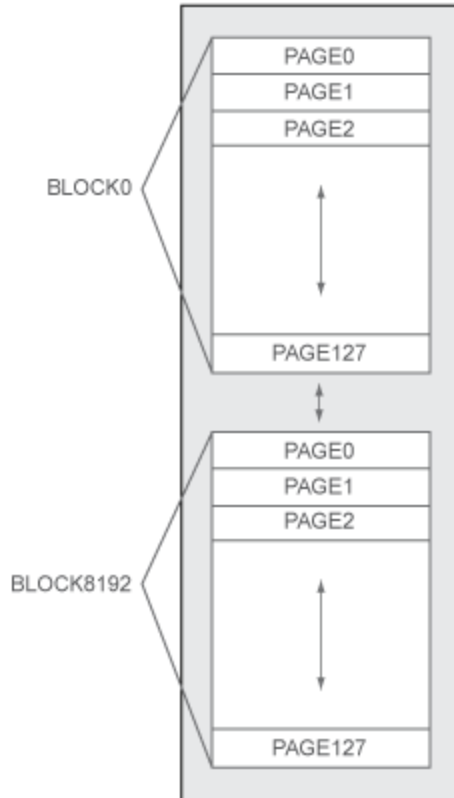
- The **ball** and the **table**.
- Quantum Mechanics.  
(quantum tunneling)
- The **electron** and the **barrier**.
- Fowler-Nordheim.  
(electron tunneling)
- Underlying process by which Flash writes (erase/program).



## Inherent disadvantages of a Flash write compared to a DRAM write?

- Can ONLY program a zero (change a 1 to 0), must be erased (set to 1) prior.
- Each write “will” require two separate Flash operations (erase/program).
- Asymmetric Flash operation (erase/program) unit sizes (Block/Page).
- Asymmetric Flash (erase/program) completion times (1.5ms/200us).
- Block/Page asymmetry (64-128X) results in RMW (Read Modify Write).
- RMW results in a write multiplicative effect called write amplification.
- Finite number of writes (erase/program) cycles (1-10K MLC/100K SLC).
- Complicated wear leveling schemes (LBA remapping) for use as an SSD.
- Writes (erase/program) will fail, requiring Bad Block Management.
- Continual performance degradation without TRIM support or Secure Erase.
- **SUMMATION:** Flash has nondeterministic and inferior write performance.

# How does a Flash write (program/erase) differ from a DRAM write?



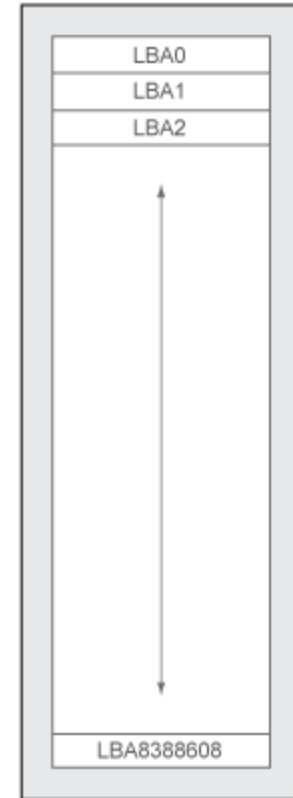
4GB FLASH SSD LAYOUT

PAGE SIZE = 4KB  
BLOCK SIZE = 512KB



4GB DRAM SSD LAYOUT

SECTOR SIZE = 512B



4GB HARD DRIVE LAYOUT

LBA SIZE = 512B

## What are common mistakes in benchmarking Flash based SSDs?

- Benchmark device immediately after purchase or Secure Erase:  
**Flash SSDs produce one-time, unsustainable, dramatically inflated results when tested “new” or after a Secure Erase.**
- Iometer Benchmark version (default settings key to IOPS):
  - Latest stable release from the official website (iometer.org) is 2006.07.27
    - IO transfer data is **pseudo random** in 2006.07.27
  - Latest release candidate or developer build at SourceForge is 2008.06.22 RC2
    - IO transfer data is “**repeating byte**” in 2008.06.22 RC2
- Running Iometer with a **contrived** IO transfer data pattern (e.g. repeating byte) will NOT provide generalized results if the device under test implements compression (SandForce 1500).

# What is an example of Iometer 2006.07.27 pseudo random data?

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	
00000000	67	01	C7	5F	9B	C4	83	A3	E4	8A	BF	56	4A	C8	3B	D5	50	2D	9D	BA	g.Ç_Ä ä ä ä VJÈ;ÖP- º
00000014	50	A8	2F	75	FA	6A	F4	4C	3F	39	36	B3	8D	5E	68	B4	8C	39	D8	39	P'/'uújôL?96º ^'h'  909
00000028	A1	37	BA	9C	F9	5B	90	15	AA	E0	5C	BD	A1	54	54	78	61	5B	11	25	i7º ù[ . .ºà\%iTTxa[. %
0000003C	A0	18	AB	C6	B3	82	43	15	53	13	12	8D	40	6D	F2	5E	D5	EE	AC	68	..«Æ³ C.S.. @mó^Öi-h
00000050	8D	38	3F	AB	63	43	CE	54	51	65	22	D7	15	37	02	51	96	D7	B2	9F	8?«cCÎTQe"x.7.Q ×²
00000064	93	A1	9A	C9	D1	20	A9	A2	59	C3	93	06	7F	1F	6B	5E	29	33	68	DD	ÈÑ @cYÄ  .k^ 3hÝ
00000078	8E	44	45	9A	C3	A5	B3	AB	CB	36	E1	61	6D	49	25	3D	92	BE	06	CB	DE Ä³³«È6áamI%='º.È
0000008C	7B	B4	4D	FB	11	A1	5D	BA	BA	4A	D3	A9	0B	8E	33	42	E2	1E	25	60	{'Mú.ij]ººJÓ@. 3Bá.º`
000000A0	B0	3F	D2	26	1A	01	1F	71	9C	39	49	2F	0E	F8	00	9B	B5	F2	D8	8D	^?Ô&...q 9I/.º.µò@
000000B4	D6	C1	E3	CC	62	21	09	BF	33	A7	82	EC	65	EE	9B	D8	B5	C6	73	B8	ÖÁäib!..¿3\$ ie @µÆs,
000000C8	B1	43	56	32	DC	8D	7F	7E	DE	E7	60	90	1C	20	36	1F	B2	08	66	2E	±CV2Ü  ~bç` . .6.º.f.
000000DC	19	80	5A	84	28	9E	4E	AE	62	DB	5E	6D	5C	09	DC	DA	0E	5C	4F	02	..ZI( N@bÛ^m\..ÛÛ.\O.
000000F0	8A	CB	61	C1	10	8C	35	72	CD	17	38	25	37	F0	94	4A	F7	7C	91	C5	ÈäÁ. 5rÍ.8%78 J+ 'Á
00000104	39	03	D6	AC	14	DC	9C	DA	9C	00	FD	43	79	F9	15	9A	8A	EE	43	0B	9.Ö-.Û Û .ýCyù.   iC.
00000118	27	6E	14	68	92	CE	97	90	D8	39	1E	1F	39	7C	65	1A	3C	EC	18	8D	'n.h'Í   09..9 e.<i.
0000012C	3B	9C	16	AC	AB	6E	7E	E1	17	3C	7F	F2	93	ED	BF	17	52	95	D7	BC	.º«m^á.< ò i ç.R ×%
00000140	6A	0A	98	EB	A9	99	D4	8C	33	36	DA	4E	0D	B2	96	2E	76	BD	5F	D6	j. è@ Ô 36ÛN.º  .v%_Ö
00000154	C6	1E	26	6B	10	3D	E3	8C	0C	CA	16	45	B5	C1	4F	3C	9E	08	24	02	Æ.&k..=ä .È.EµÁO< .º.
00000168	7A	18	07	9A	D5	4B	00	EE	B3	88	1F	69	F5	AA	EA	D0	3B	BC	56	16	z.. ÖK.i³  .içºèD;¼V.
0000017C	6D	3E	AF	09	0F	CA	5A	E0	83	DE	88	54	2A	E8	9D	35	06	F9	75	23	m>~..ÈZà P T*è 5.ùu#
00000190	14	58	03	06	4D	57	25	9B	0A	03	80	52	8B	73	39	DC	25	2D	4D	1E	.X..MW%  .. R s9Û%-M.
000001A4	7E	7E	F6	5E	61	BB	20	4E	29	6D	3F	08	70	59	B3	20	08	A5	92	1B	~º^a» N)m?..pY³.º'.
000001B8	89	F7	BC	CB	E4	E9	63	AA	B4	AC	98	77	5A	E6	EC	1E	66	70	A1	37	+¼Èäéçº^~ wZei.fpi7
000001CC	AD	13	3B	BD	4F	EB	5D	38	EA	49	C5	79	F8	B1	2B	CE	55	83	D1	F3	-.;%Oè 8èIÁyat+ÍU Ñó
000001E0	98	03	01	38	80	F6	7D	5E	60	96	66	A6	BF	74	54	CC	54	4B	12	8F	.8 ö ^` f çtTITK.
000001F4	08	85	06	5A	AD	63	A7	F5	9F	8A	19	AB	F2	09	46	E1	F9	81	BC	01	.Z-cSÖ  .«ò.Fáù ¼.
00000208	4B	51	5E	E4	54	66	22	82	84	DB	16	FD	76	87	6B	1E	64	90	09	29	KQ^ätf"  Û.ýv k.d  .
0000021C	50	00	62	9A	1A	39	2D	CE	2F	74	A5	CA	CF	AD	82	73	E8	FB	06	47	P.b . 9-Í/t#ÈÍ- sèù.G
00000230	7D	AF	D9	8F	98	47	76	52	83	80	9C	DF	C1	F2	A7	70	F1	86	85	7B	}~Û  GvR   BÁòSpñ  {

# What is an example of lometer 2008.06.22 RC2 “repeating byte”?

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13		
00000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000014	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000028	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0000003C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000064	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000078	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0000008C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000B4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000C8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000DC	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000104	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000118	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0000012C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000154	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000168	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0000017C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001A4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001B8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001CC	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001F4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0000208	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000021C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0000230	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

# What is an example of lometer 2008.06.22 RC2 “repeating byte”?

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13		
00000000	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
00000014	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
00000028	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
0000003C	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
00000050	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
00000064	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
00000078	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
0000008C	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
000000A0	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
000000B4	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
000000C8	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
000000DC	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
000000F0	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
00000104	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
00000118	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
0000012C	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
00000140	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
00000154	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
00000168	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
0000017C	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
00000190	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
000001A4	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
000001B8	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
000001CC	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
000001E0	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
000001F4	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
0000208	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
000021C	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
0000230	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....

# What is an example of lometer 2008.06.22 RC2 “repeating byte”?

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13		
00000000	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
00000014	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
00000028	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
0000003C	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
00000050	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
00000064	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
00000078	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
0000008C	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
000000A0	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
000000B4	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
000000C8	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
000000DC	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
000000F0	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
00000104	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
00000118	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
0000012C	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
00000140	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
00000154	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
00000168	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
0000017C	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
00000190	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
000001A4	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
000001B8	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
000001CC	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
000001E0	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
000001F4	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
00000208	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
0000021C	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
00000230	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####

## Why is the Iometer IO transfer data so critical to IOPS benchmarking?

- Iometer 2008.06.22 RC write IO uses a “repeating byte” data pattern (examples shown prior were 0x00, 0x01, 0x23) which is copied 4096 times to completely fill the 4KB transfer buffer.
- Benchmarking using a contrived data pattern (repeating byte) will not provide generalized results with some Flash SSDs, thus are not indicative of generalized ZIL Accelerator ability.
- Both the OCZ Vertex 2 EX and the OCZ Vertex Pro SSD use the SandForce 1500 controller which implements compression at the drive level. Benchmarking with an Iometer version that defaults to using extremely compressible data will, in these cases, show dramatically inflated write IOPS results.
- All results shown in this presentation use Pseudo Random.

# Iometer benchmark devices, distributions, and procedure:

## Benchmark Devices Under Test:

- OCZ Vertex 2 EX (SLC Flash SSD)
- OCZ Vertex 2 Pro (MLC Flash SSD)
- DDRdrive X1 (DRAM SSD)



## Benchmark 4KB Write IOPS Distributions:

- 100% Sequential
- 50% Sequential / 50% Random
- 100% Random



## Benchmark Procedure :

- Secure Erase (SE) Flash SSD (no SE with X1)
- Start test, run 60 minutes (see slide 1 of 2).
- Stop test, allow device/host to be quiescent (no tests or host activity) for 60 minutes.
- Restart test (secure erase not repeated) for a second 60 minute run (see slide 2 of 2).



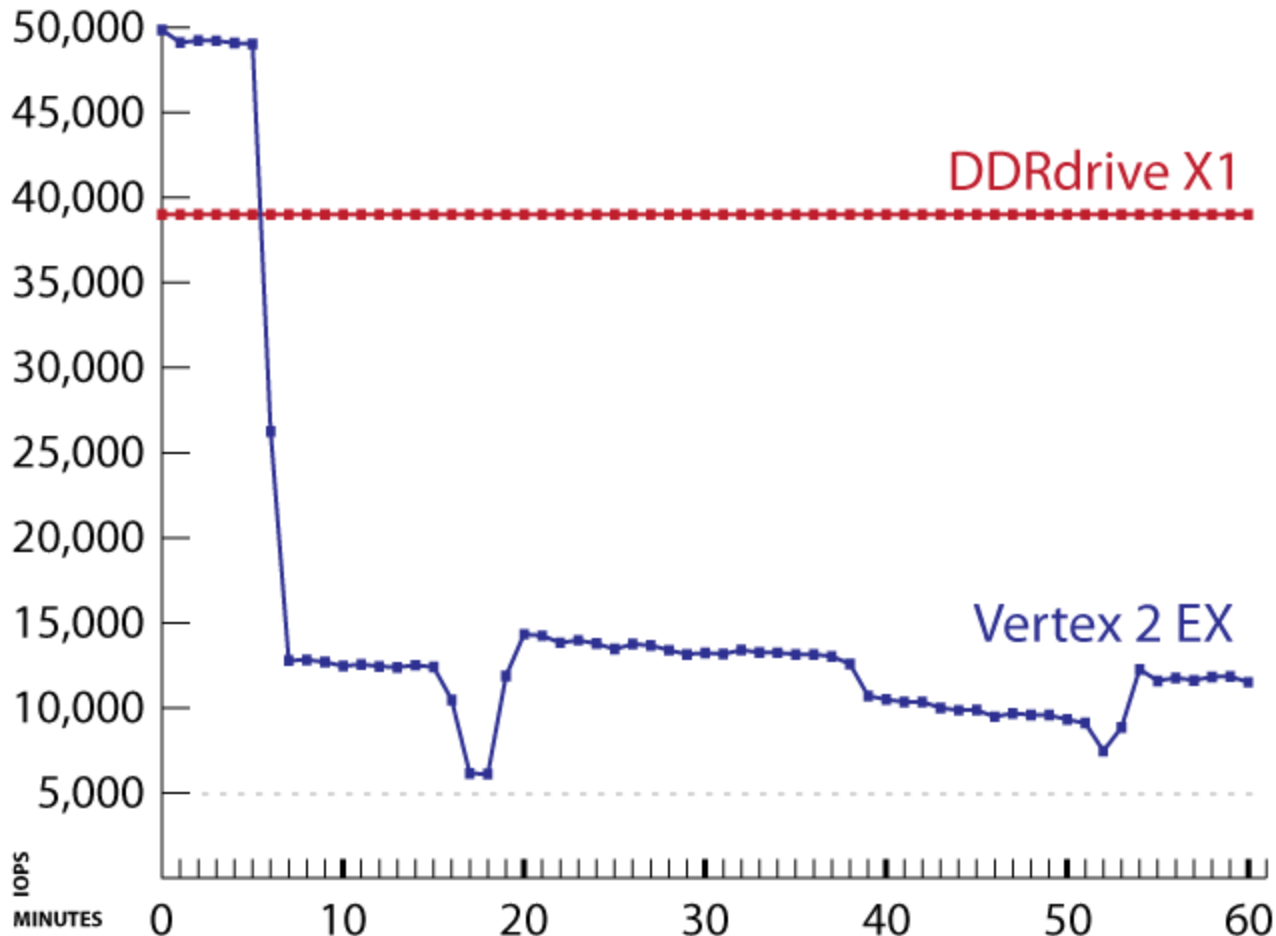
# The untold truth about Flash SSD Write IOPS degradation?

[Product Packaging]

4KB Random Write:  
Up to 50,000 IOPS



# The untold truth about Flash SSD Write IOPS degradation?



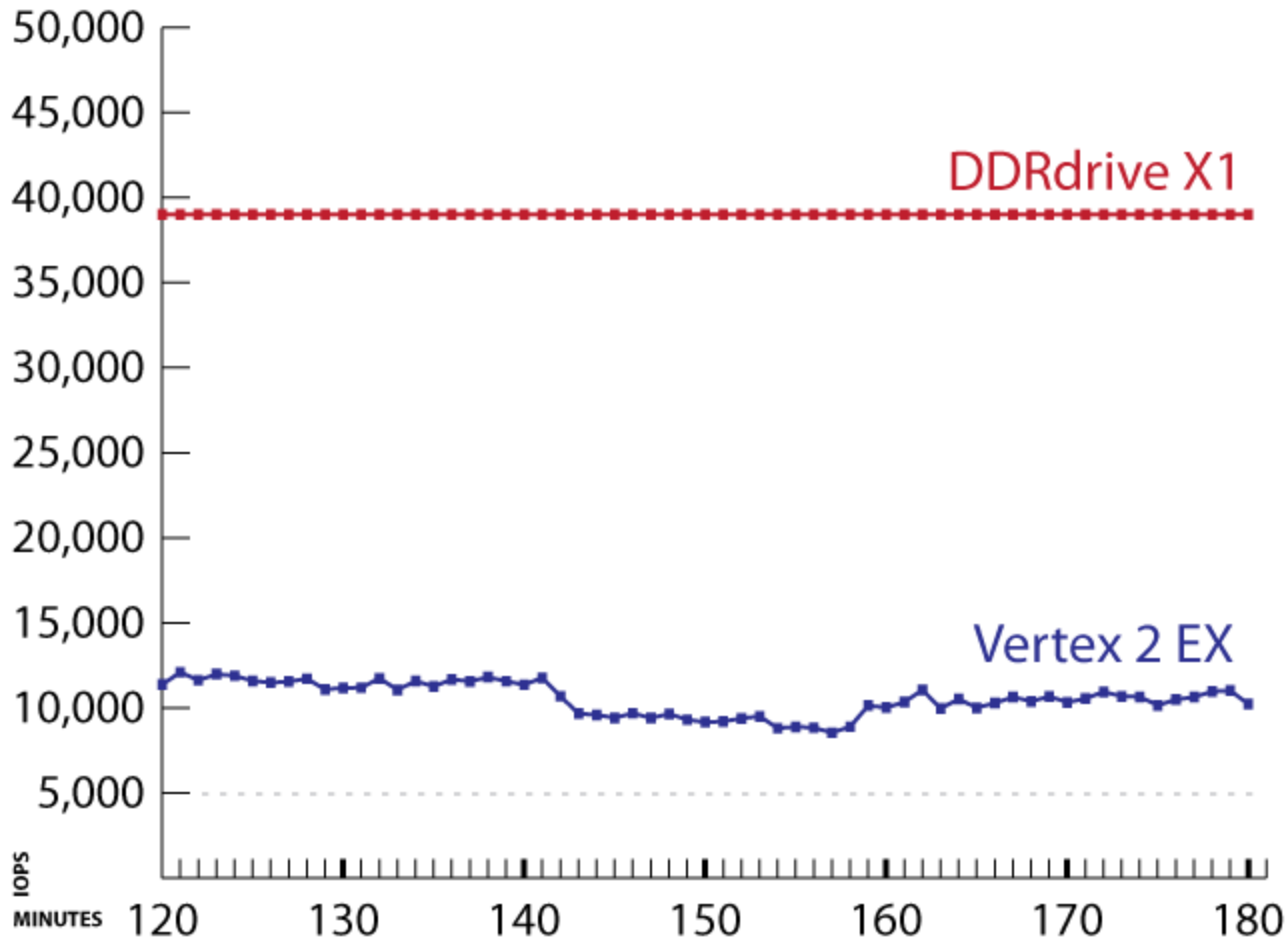
4KB Sequential Writes

Slide 1 of 2.

**SECURE ERASE**  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

IOmeter 2006.07.27  
IO Alignment = 4KB  
Transfer Size = 4KB  
Transfer Data = Pseudo Random  
100% Sequential Distribution  
100% Write Distribution  
Queue Depth = 32  
Disk Workers = CPU (2)  
Target Disk = PhysicalDrive  
Time 0 = Start of Test (1 sec)  
Time 1+ = Last Update (60 sec)  
TRIM Not Passed to Devices  
Intel ICH10R Chipset  
Intel RST 9.6.0.1014 Driver  
OCZ Vertex 2 EX 50GB FW 1.11

# The untold truth about Flash SSD Write IOPS degradation?



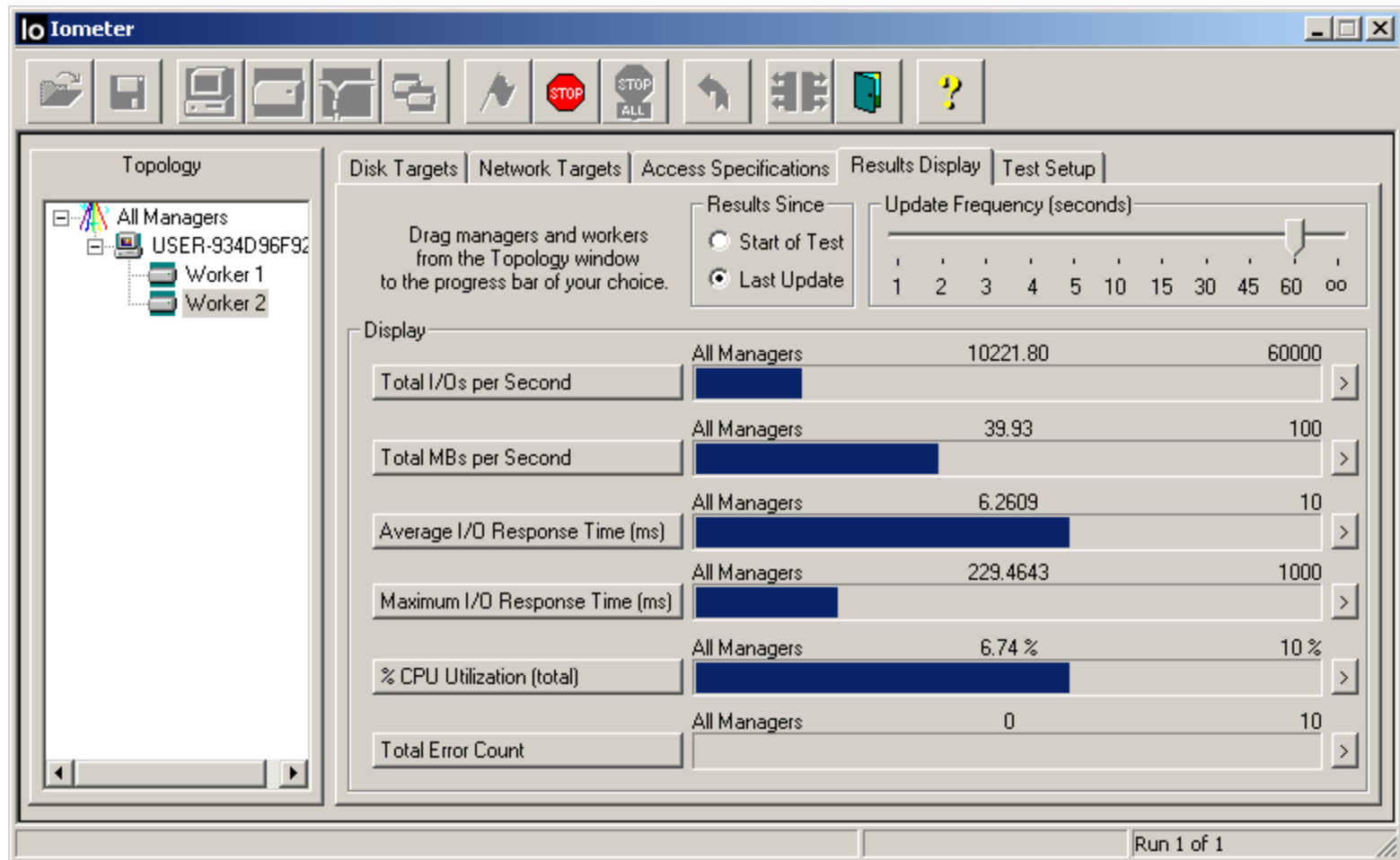
4KB Sequential Writes

Slide 2 of 2.

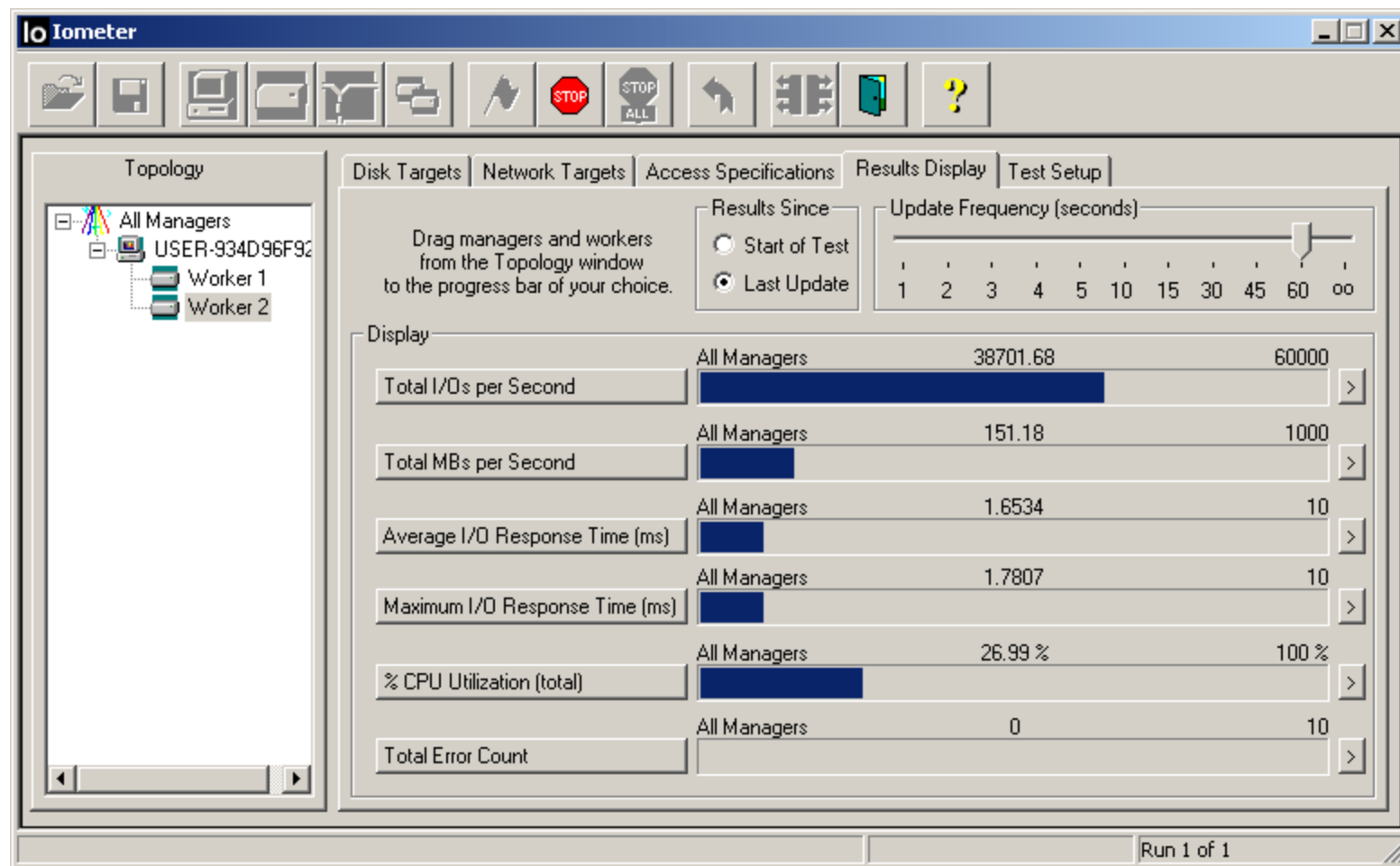
SECURE ERASE  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

IOmeter 2006.07.27  
IO Alignment = 4KB  
Transfer Size = 4KB  
Transfer Data = Pseudo Random  
100% Sequential Distribution  
100% Write Distribution  
Queue Depth = 32  
Disk Workers = CPU (2)  
Target Disk = PhysicalDrive  
Time 0 = Start of Test (1 sec)  
Time 1+ = Last Update (60 sec)  
TRIM Not Passed to Devices  
Intel ICH10R Chipset  
Intel RST 9.6.0.1014 Driver  
OCZ Vertex 2 EX 50GB FW 1.11

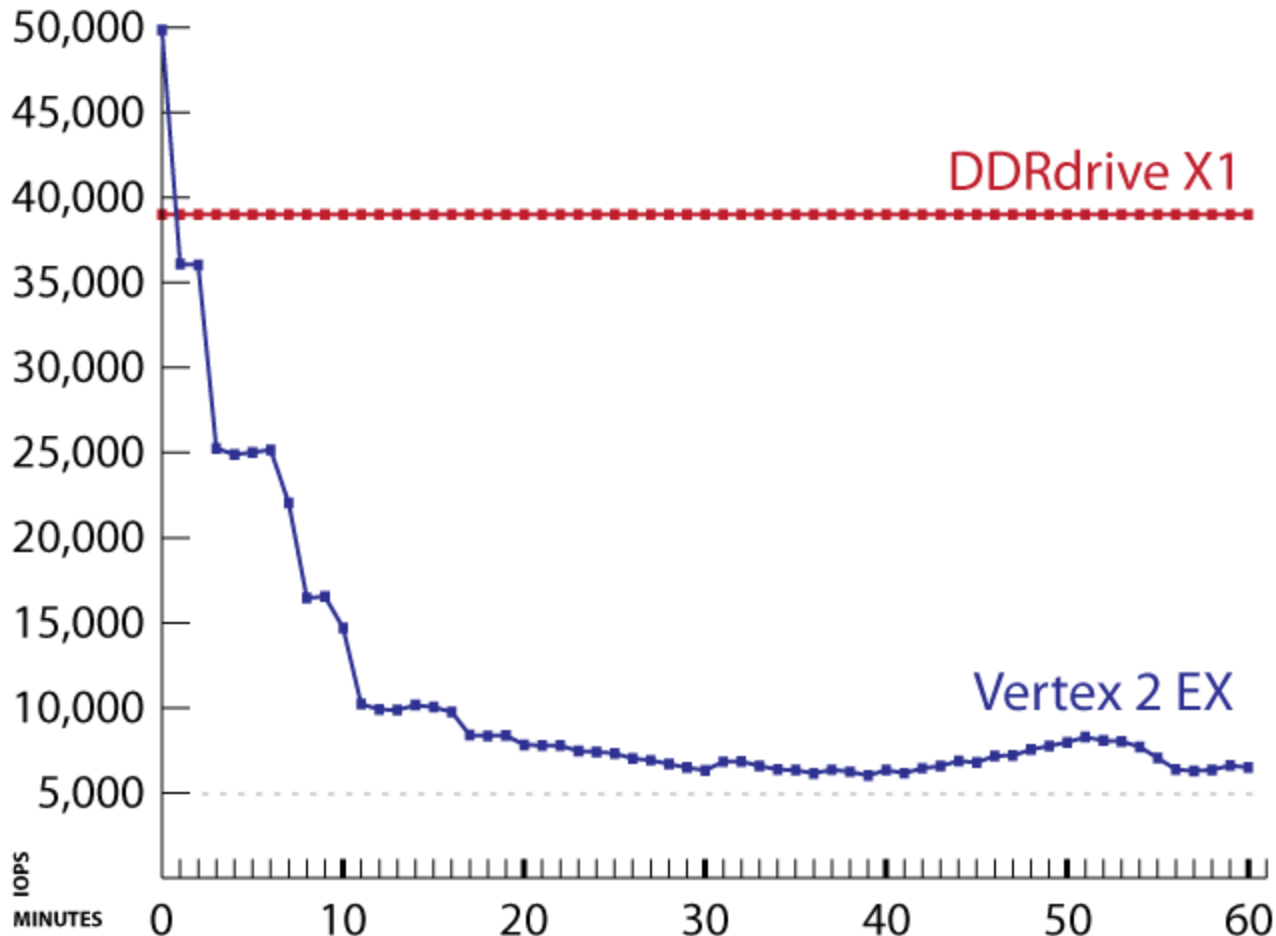
# Vertex 2 EX 100% Sequential 4KB Write 4KB Aligned 180 Minute Screenshot:



# DDRdrive X1 100% Sequential 4KB Write 4KB Aligned 180 Minute Screenshot:



# The untold truth about Flash SSD Write IOPS degradation?



4KB Mixed Writes

Slide 1 of 2.

**SECURE ERASE**  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

IOmeter 2006.07.27

IO Alignment = 4KB

Transfer Size = 4KB

Transfer Data = Pseudo Random

50% Sequential Distribution

50% Random Distribution

100% Write Distribution

Queue Depth = 32

Disk Workers = CPU (2)

Target Disk = PhysicalDrive

Time 0 = Start of Test (1 sec)

Time 1+ = Last Update (60 sec)

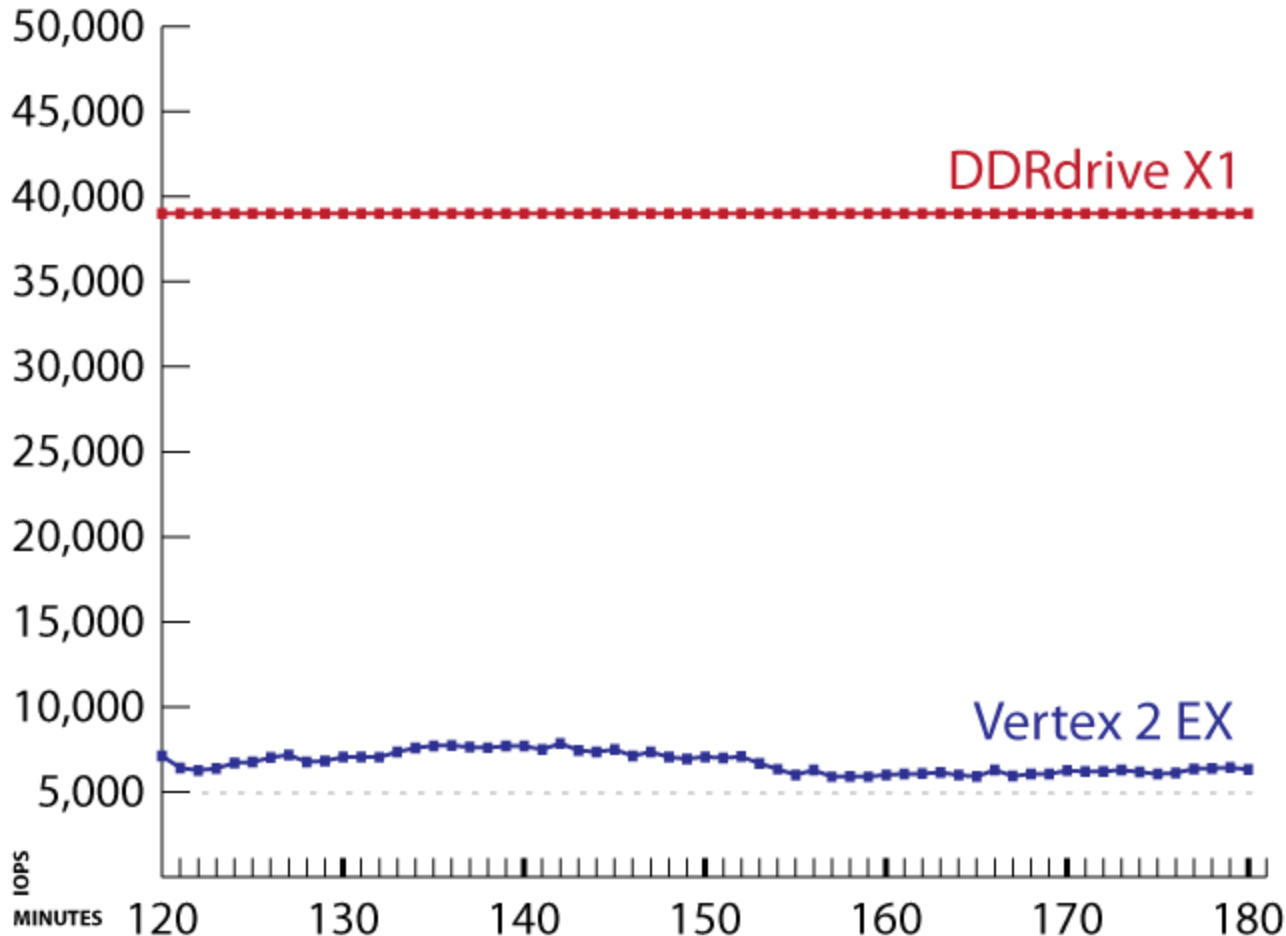
TRIM Not Passed to Devices

Intel ICH10R Chipset

Intel RST 9.6.0.1014 Driver

OCZ Vertex 2 EX 50GB FW 1.11

# The untold truth about Flash SSD Write IOPS degradation?



4KB Mixed Writes

Slide 2 of 2.

SECURE ERASE  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

IOmeter 2006.07.27

IO Alignment = 4KB

Transfer Size = 4KB

Transfer Data = Pseudo Random

50% Sequential Distribution

50% Random Distribution

100% Write Distribution

Queue Depth = 32

Disk Workers = CPU (2)

Target Disk = PhysicalDrive

Time 0 = Start of Test (1 sec)

Time 1+ = Last Update (60 sec)

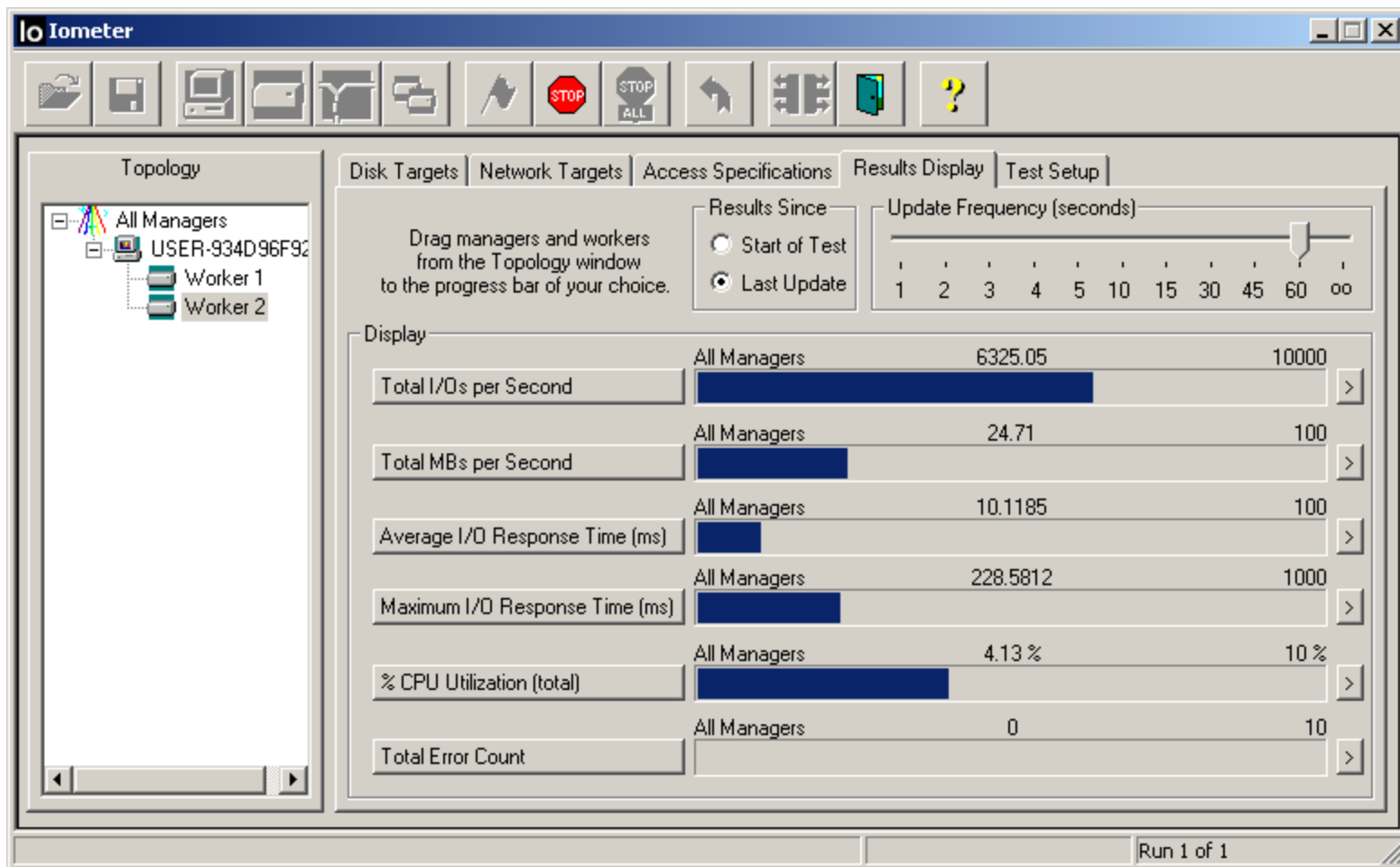
TRIM Not Passed to Devices

Intel ICH10R Chipset

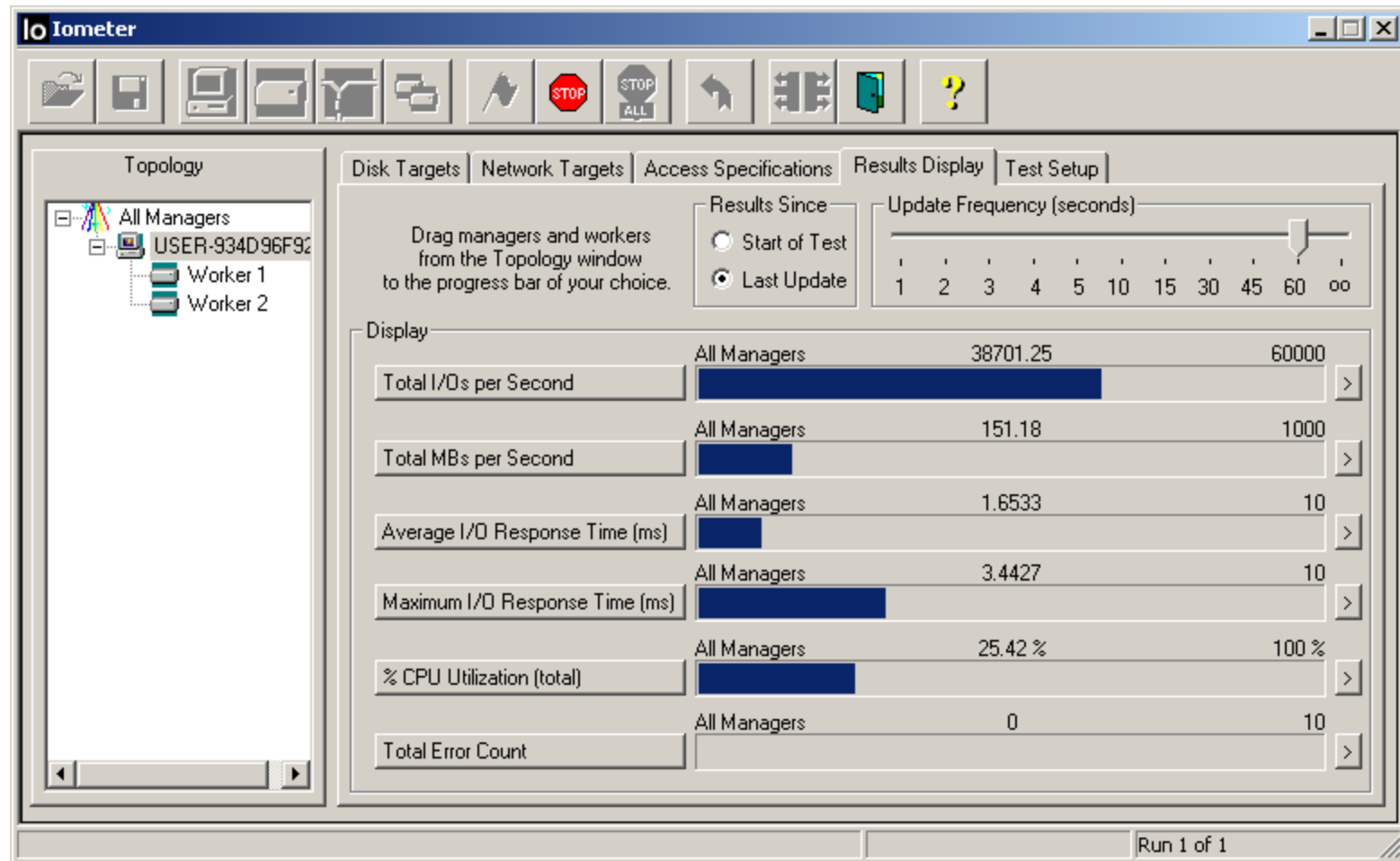
Intel RST 9.6.0.1014 Driver

OCZ Vertex 2 EX 50GB FW 1.11

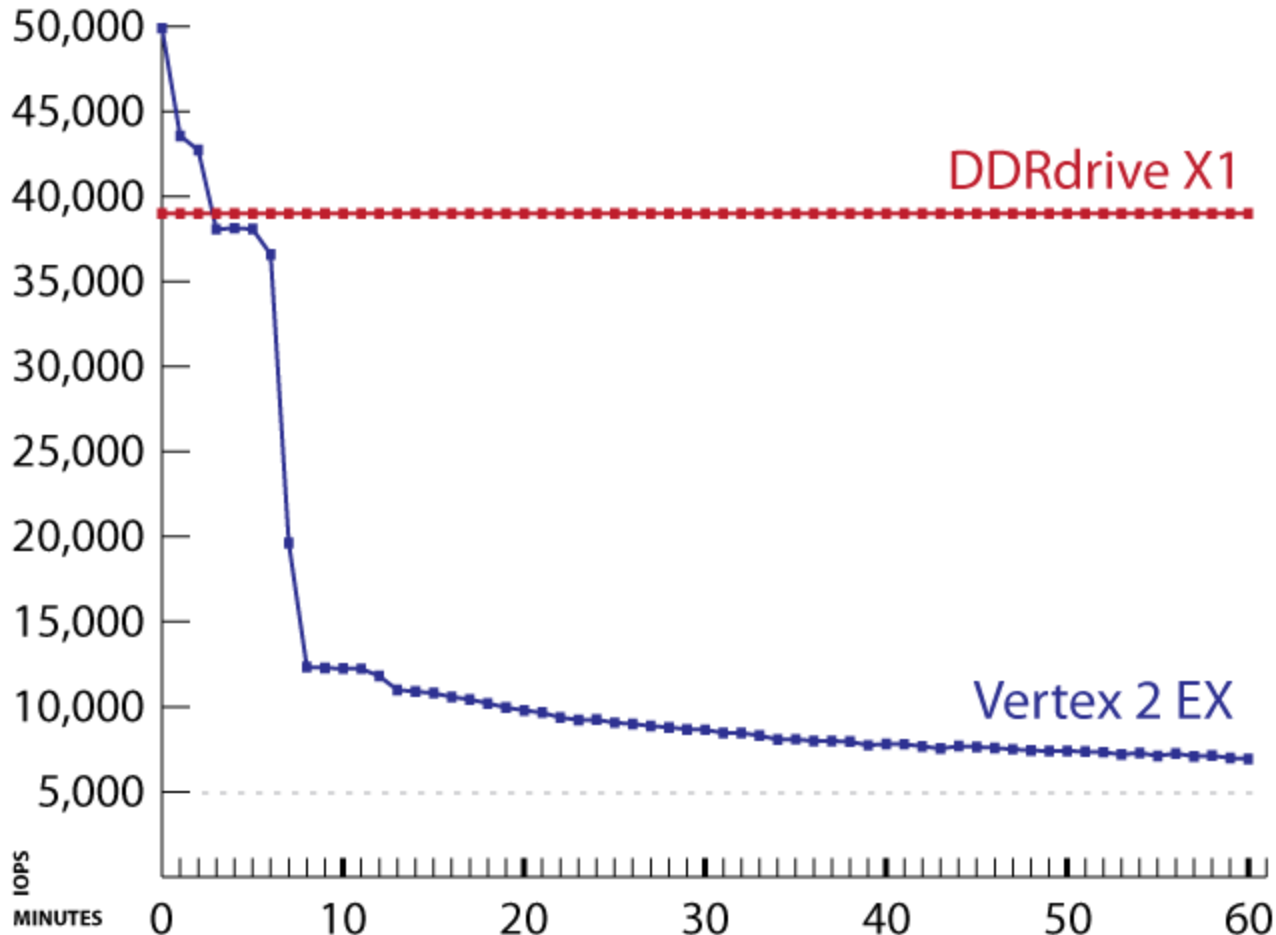
# Vertex 2 EX 50% Seq./50% Ran. 4KB Write 4KB Aligned 180 Minute Screenshot:



# DDRdrive X1 50% Seq./50% Ran. 4KB Write 4KB Aligned 180 Minute Screenshot:



# The untold truth about Flash SSD Write IOPS degradation?



4KB Random Writes

Slide 1 of 2.

**SECURE ERASE**  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

IOmeter 2006.07.27

IO Alignment = 4KB

Transfer Size = 4KB

Transfer Data = Pseudo Random

100% Random Distribution

100% Write Distribution

Queue Depth = 32

Disk Workers = CPU (2)

Target Disk = PhysicalDrive

Time 0 = Start of Test (1 sec)

Time 1+ = Last Update (60 sec)

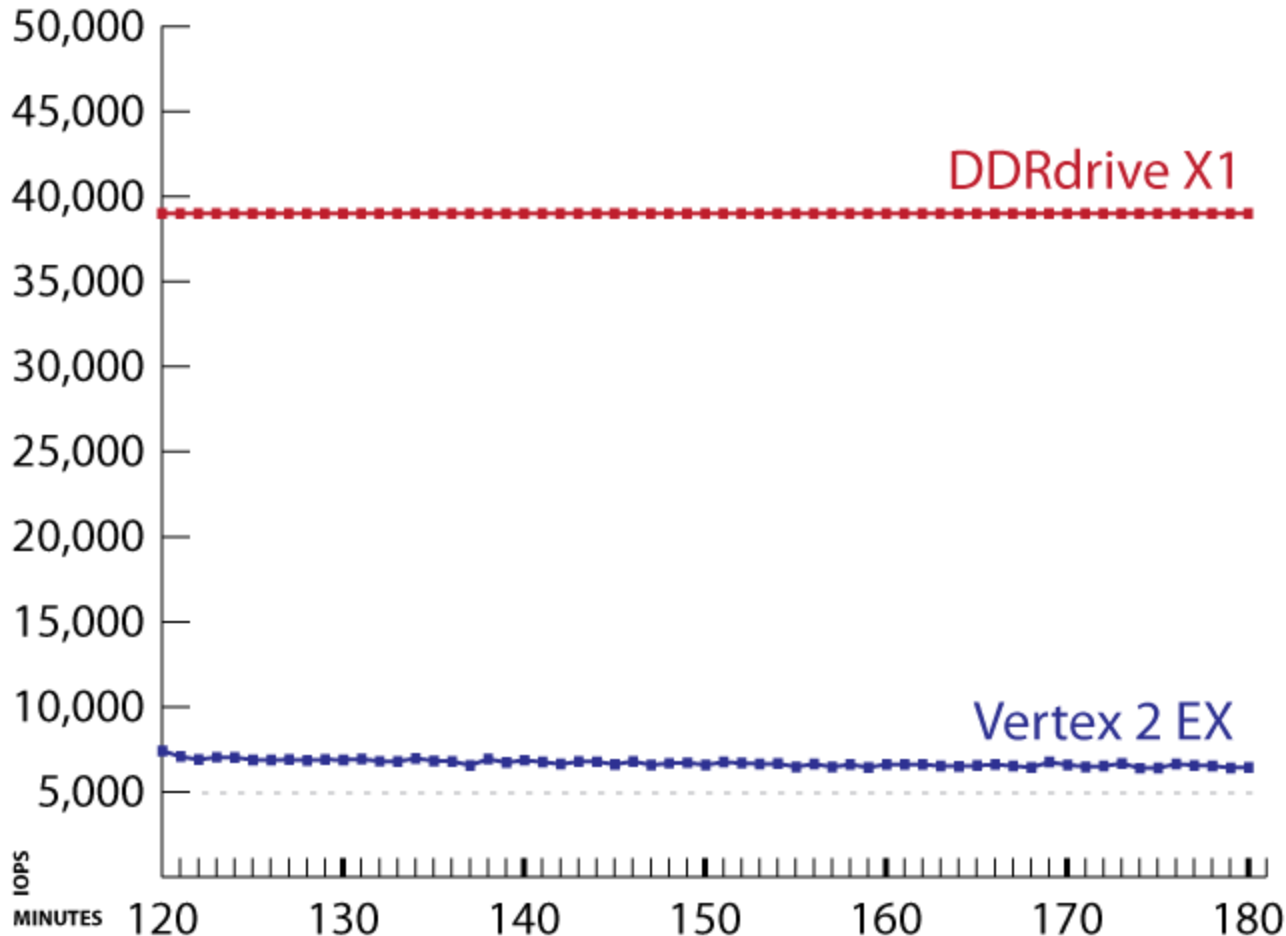
TRIM Not Passed to Devices

Intel ICH10R Chipset

Intel RST 9.6.0.1014 Driver

OCZ Vertex 2 EX 50GB FW 1.11

# The untold truth about Flash SSD Write IOPS degradation?



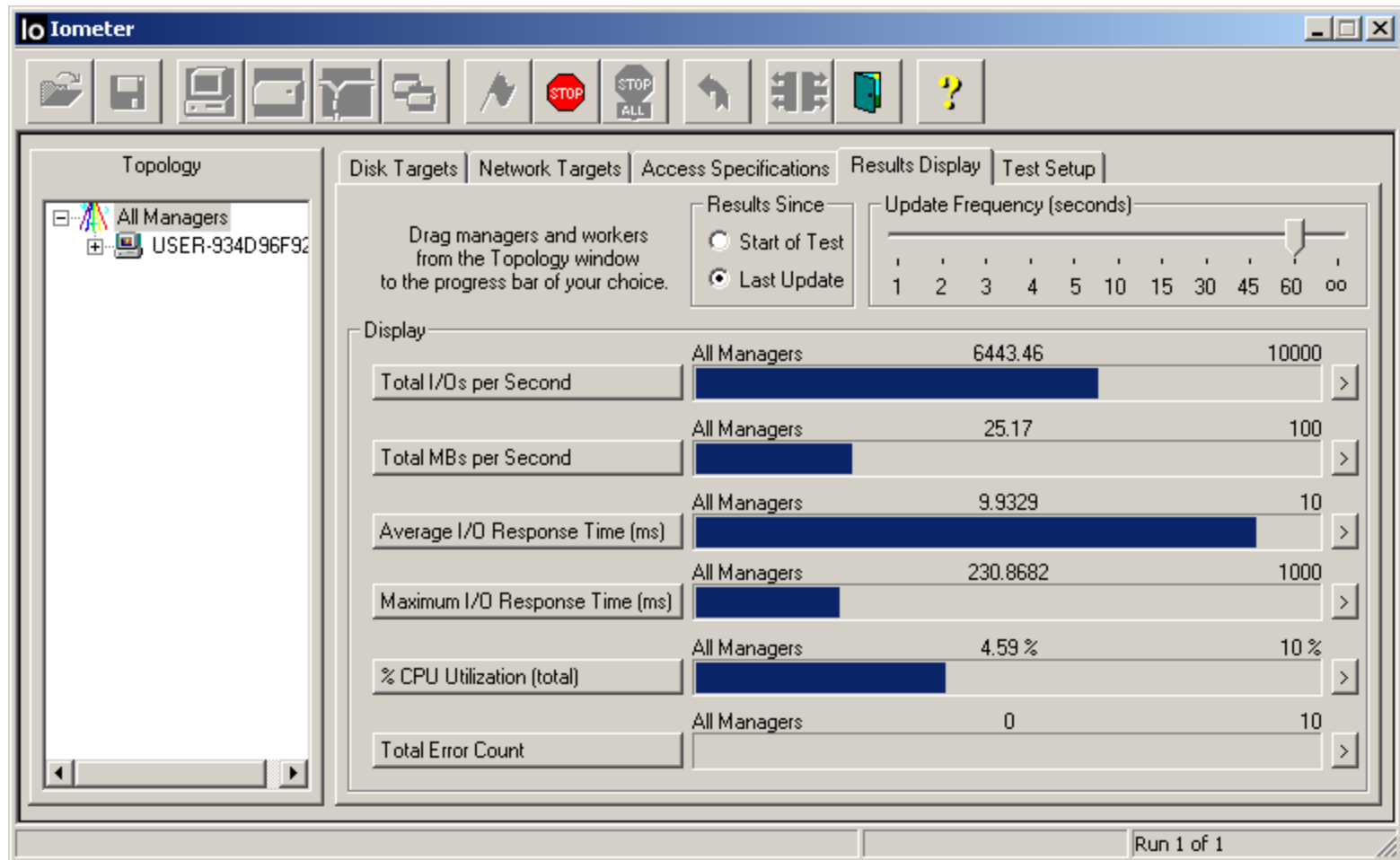
4KB Random Writes

Slide 2 of 2.

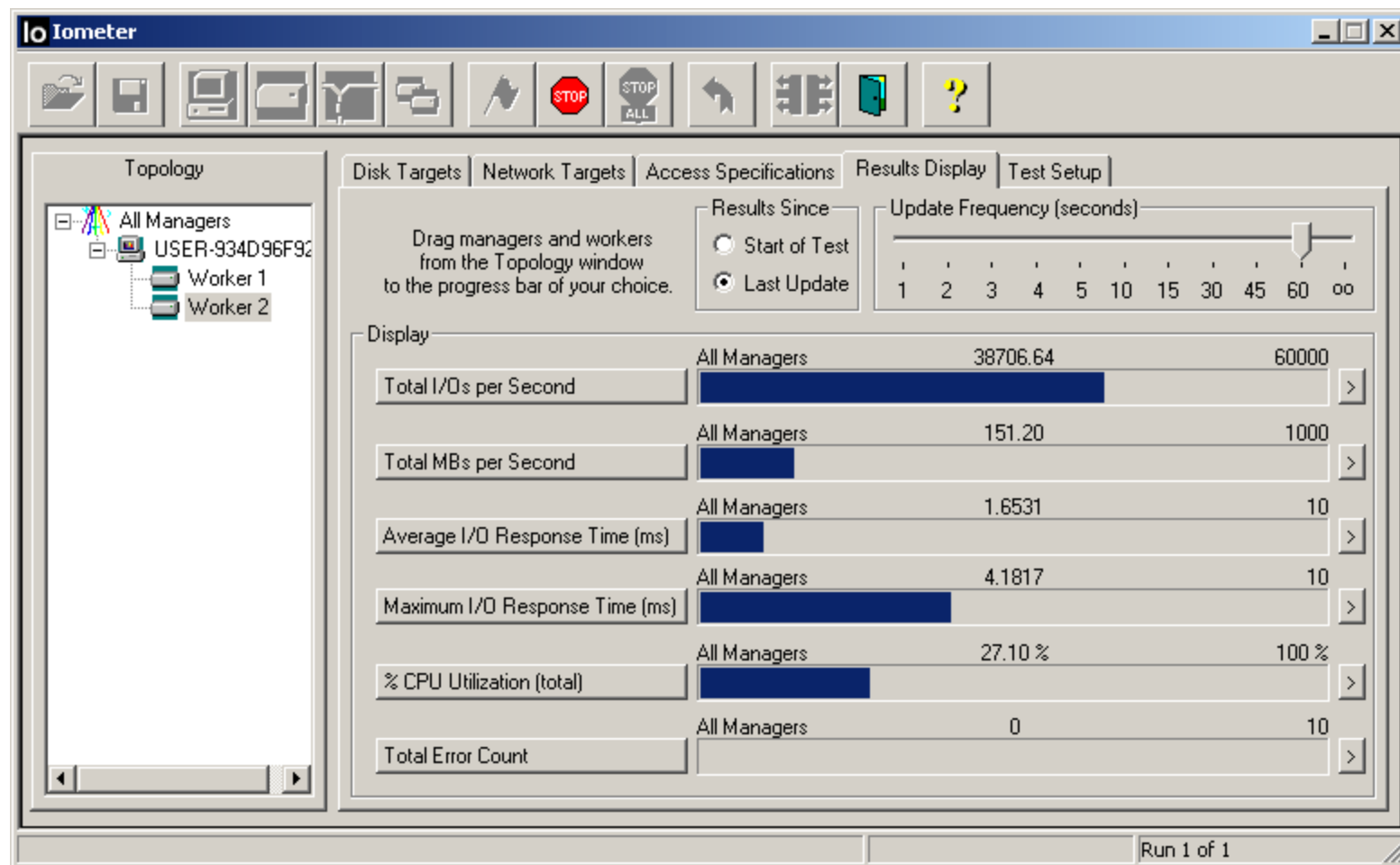
SECURE ERASE  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

IOmeter 2006.07.27  
IO Alignment = 4KB  
Transfer Size = 4KB  
Transfer Data = Pseudo Random  
100% Random Distribution  
100% Write Distribution  
Queue Depth = 32  
Disk Workers = CPU (2)  
Target Disk = PhysicalDrive  
Time 0 = Start of Test (1 sec)  
Time 1+ = Last Update (60 sec)  
TRIM Not Passed to Devices  
Intel ICH10R Chipset  
Intel RST 9.6.0.1014 Driver  
OCZ Vertex 2 EX 50GB FW 1.11

# Vertex 2 EX 100% Random 4KB Write 4KB Aligned 180 Minute Screenshot:



# DDRdrive X1 100% Random 4KB Write 4KB Aligned 180 Minute Screenshot:



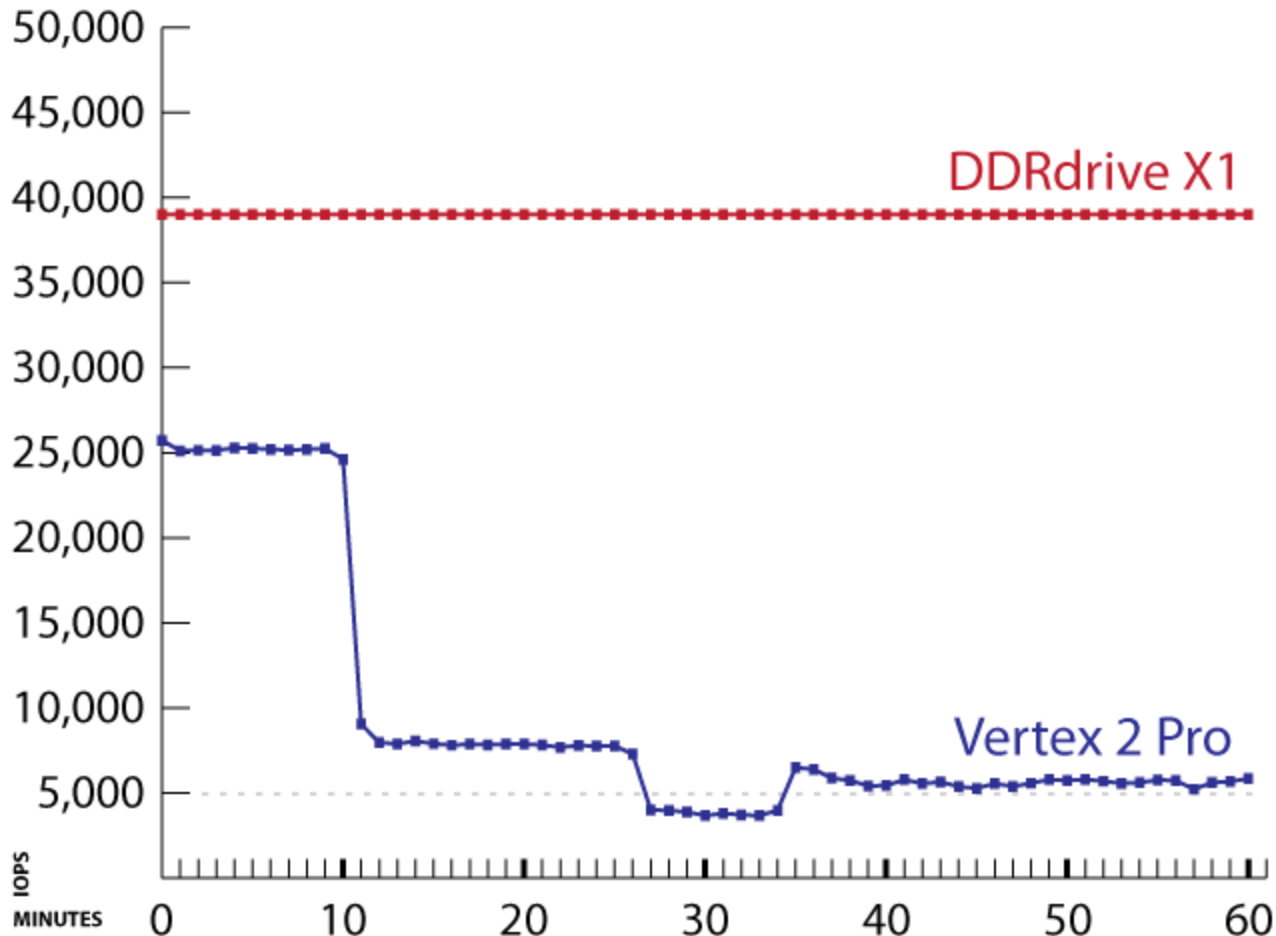
# The untold truth about Flash SSD Write IOPS degradation?

[Product Packaging]

4KB Random Write:  
Up to 50,000 IOPS



# The untold truth about Flash SSD Write IOPS degradation?



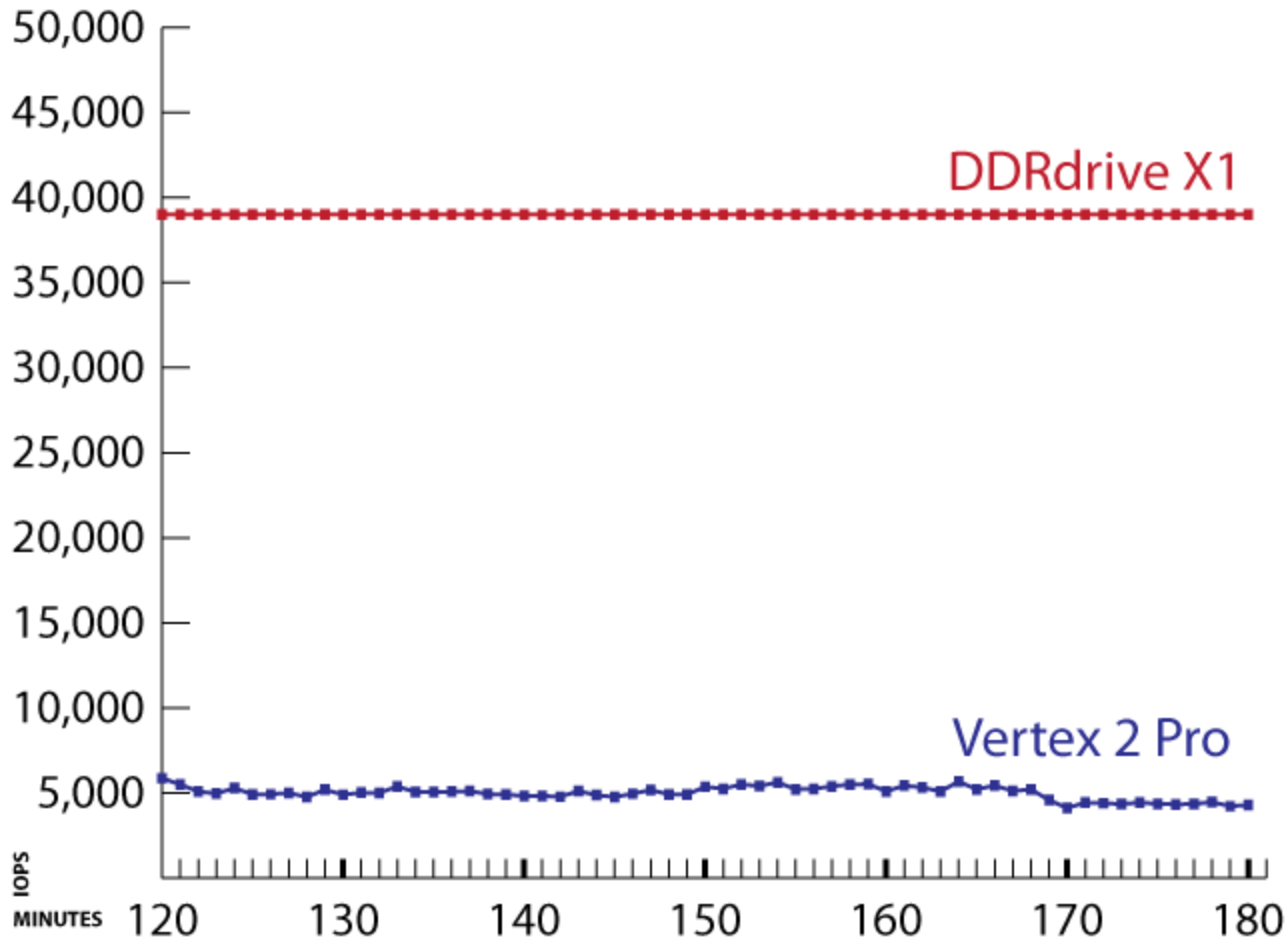
4KB Sequential Writes

Slide 1 of 2.

**SECURE ERASE**  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

IOmeter 2006.07.27  
IO Alignment = 4KB  
Transfer Size = 4KB  
Transfer Data = Pseudo Random  
100% Sequential Distribution  
100% Write Distribution  
Queue Depth = 32  
Disk Workers = CPU (2)  
Target Disk = PhysicalDrive  
Time 0 = Start of Test (1 sec)  
Time 1+ = Last Update (60 sec)  
TRIM Not Passed to Devices  
Intel ICH10R Chipset  
Intel RST 9.6.0.1014 Driver  
OCZ Vertex 2 Pro 50GB FW 1.11

# The untold truth about Flash SSD Write IOPS degradation?



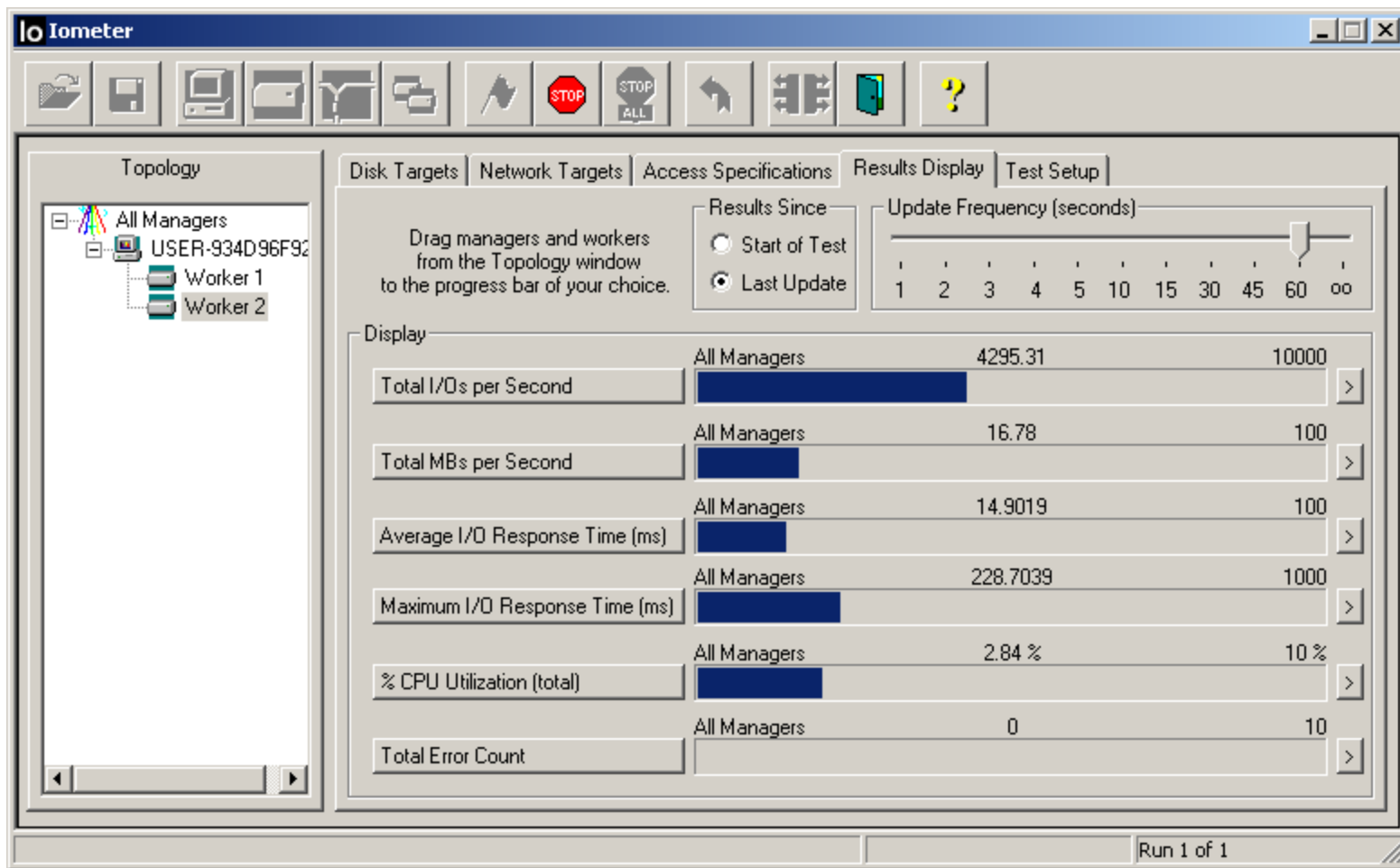
4KB Sequential Writes

Slide 2 of 2.

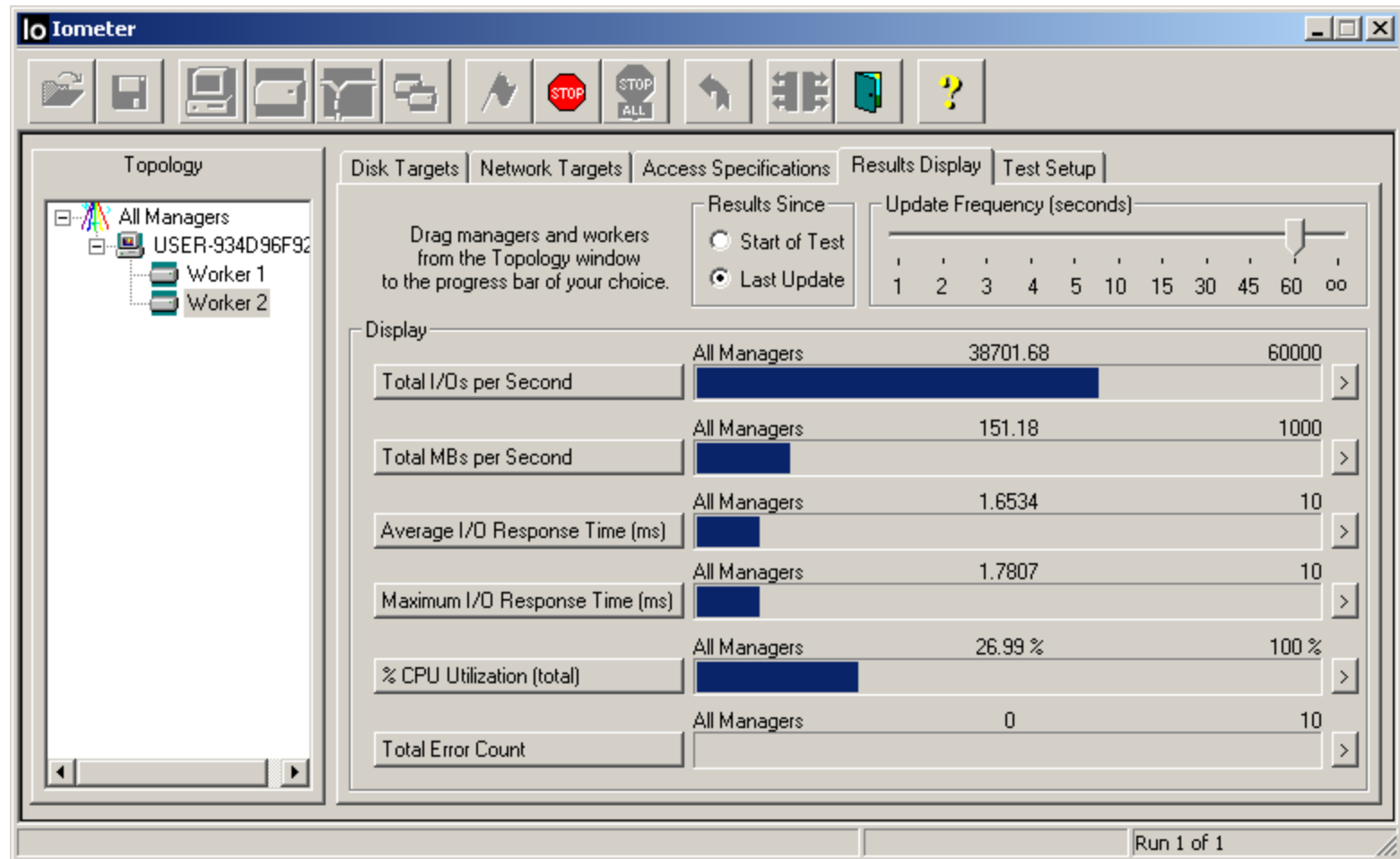
SECURE ERASE  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

IOmeter 2006.07.27  
IO Alignment = 4KB  
Transfer Size = 4KB  
Transfer Data = Pseudo Random  
100% Sequential Distribution  
100% Write Distribution  
Queue Depth = 32  
Disk Workers = CPU (2)  
Target Disk = PhysicalDrive  
Time 0 = Start of Test (1 sec)  
Time 1+ = Last Update (60 sec)  
TRIM Not Passed to Devices  
Intel ICH10R Chipset  
Intel RST 9.6.0.1014 Driver  
OCZ Vertex 2 Pro 50GB FW 1.11

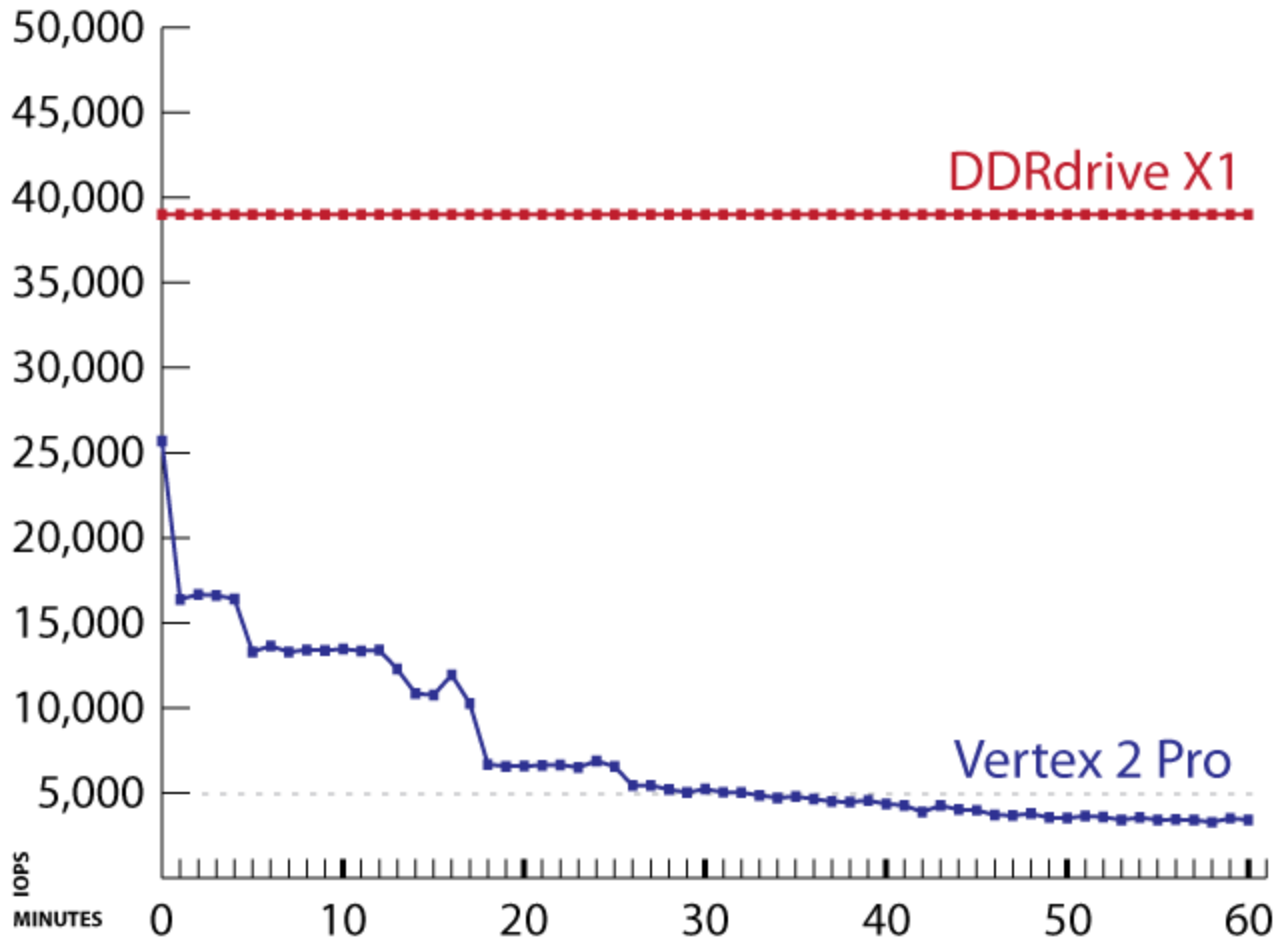
# Vertex 2 Pro 100% Sequential 4KB Write 4KB Aligned 180 Minute Screenshot:



# DDRdrive X1 100% Sequential 4KB Write 4KB Aligned 180 Minute Screenshot:



# The untold truth about Flash SSD Write IOPS degradation?



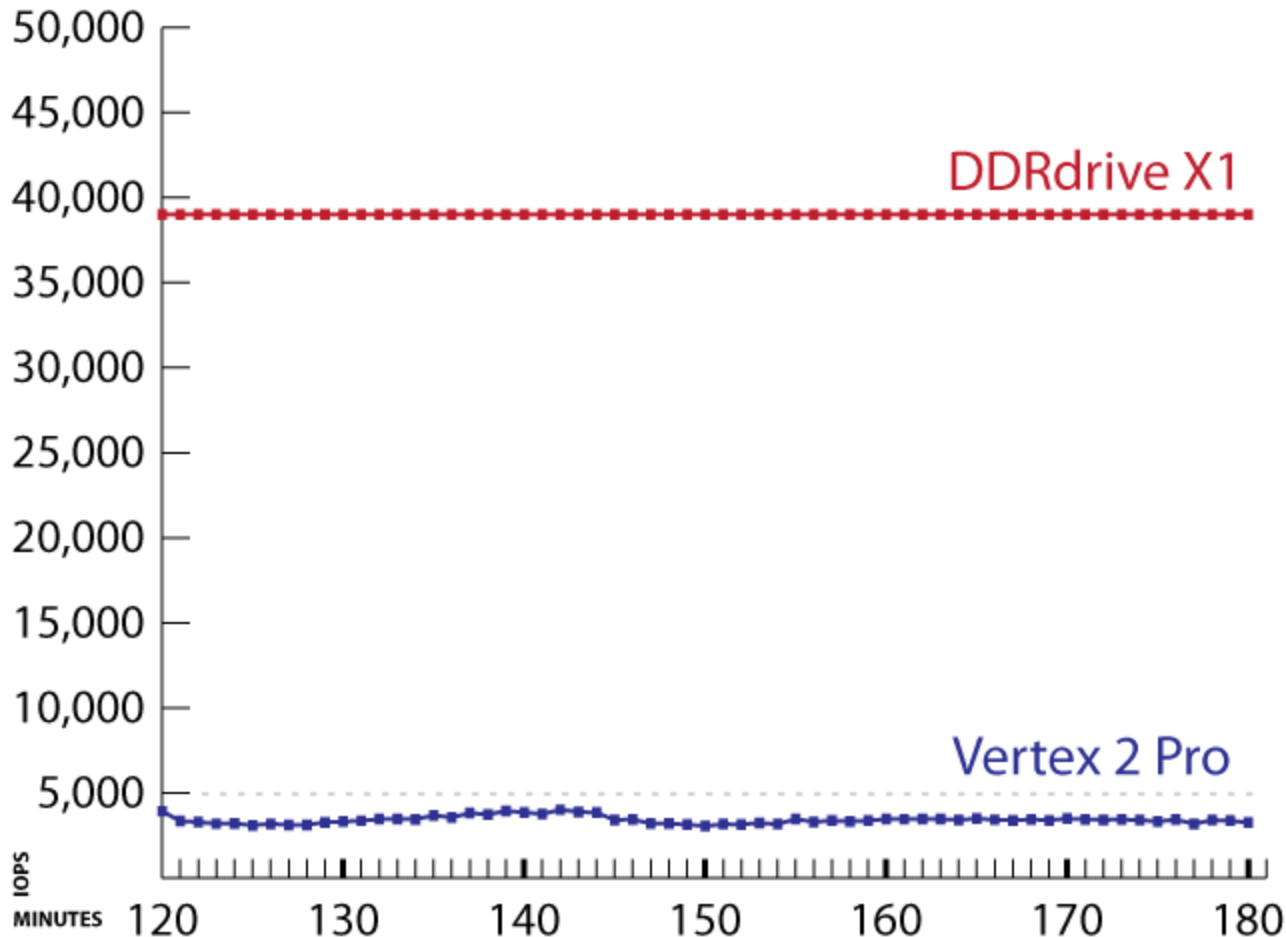
4KB Mixed Writes

Slide 1 of 2.

**SECURE ERASE**  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

IOmeter 2006.07.27  
IO Alignment = 4KB  
Transfer Size = 4KB  
Transfer Data = Pseudo Random  
50% Sequential Distribution  
50% Random Distribution  
100% Write Distribution  
Queue Depth = 32  
Disk Workers = CPU (2)  
Target Disk = PhysicalDrive  
Time 0 = Start of Test (1 sec)  
Time 1+ = Last Update (60 sec)  
TRIM Not Passed to Devices  
Intel ICH10R Chipset  
Intel RST 9.6.0.1014 Driver  
OCZ Vertex 2 Pro 50GB FW 1.11

# The untold truth about Flash SSD Write IOPS degradation?



4KB Mixed Writes

Slide 2 of 2.

SECURE ERASE  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

IOmeter 2006.07.27

IO Alignment = 4KB

Transfer Size = 4KB

Transfer Data = Pseudo Random

50% Sequential Distribution

50% Random Distribution

100% Write Distribution

Queue Depth = 32

Disk Workers = CPU (2)

Target Disk = PhysicalDrive

Time 0 = Start of Test (1 sec)

Time 1+ = Last Update (60 sec)

TRIM Not Passed to Devices

Intel ICH10R Chipset

Intel RST 9.6.0.1014 Driver

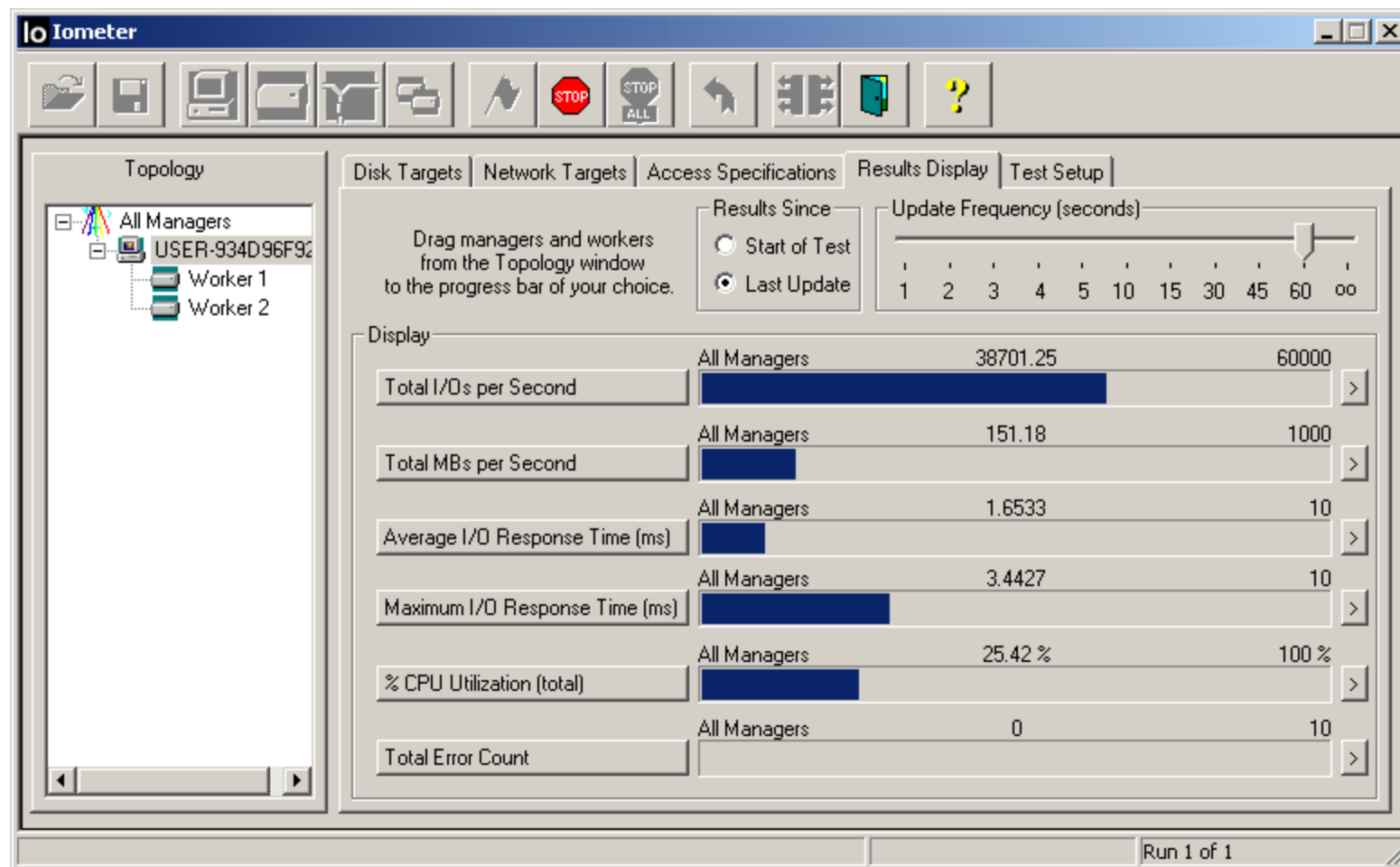
OCZ Vertex 2 Pro 50GB FW 1.11

# Vertex 2 Pro 50% Seq./50% Ran. 4KB Write 4KB Aligned 180 Minute Screenshot:

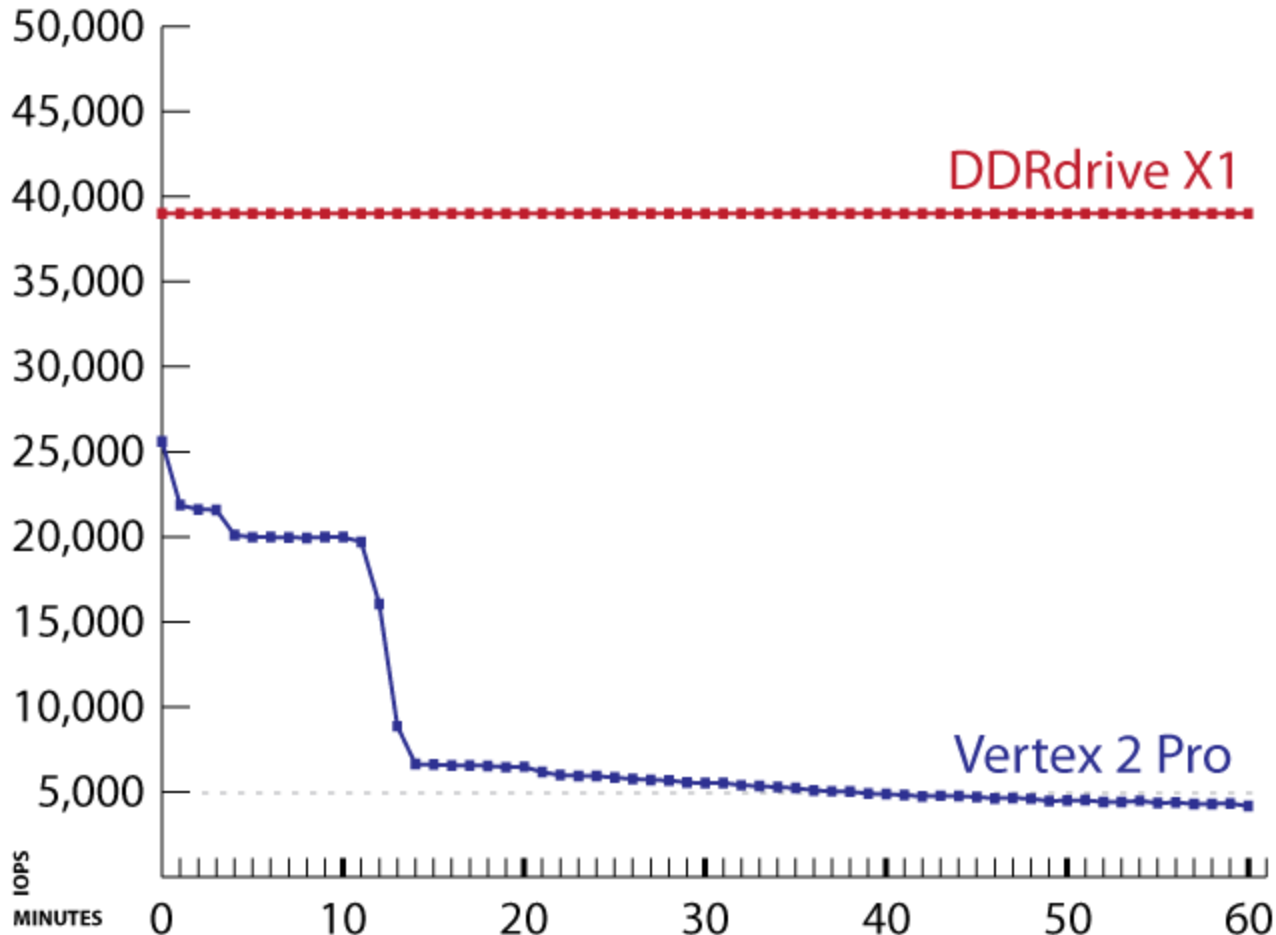
The screenshot shows the Iometer software interface. On the left, the 'Topology' window displays a tree structure with 'All Managers' expanded to show 'USER-934D96F92', 'Worker 1', and 'Worker 2'. The main area is divided into tabs: 'Disk Targets', 'Network Targets', 'Access Specifications', 'Results Display', and 'Test Setup'. The 'Results Display' tab is selected, showing a table of performance metrics for 'All Managers'. The table includes columns for the metric name, the current value, and a target value. A progress bar is visible for each metric, and a 'Results Since' dropdown is set to 'Last Update'. An 'Update Frequency (seconds)' slider is set to 60. The status bar at the bottom right indicates 'Run 1 of 1'.

Metric	Value	Target
Total I/Os per Second	3252.45	10000
Total MBs per Second	12.70	100
Average I/O Response Time (ms)	19.6793	100
Maximum I/O Response Time (ms)	281.5113	1000
% CPU Utilization (total)	2.15 %	10 %
Total Error Count	0	10

# DDRdrive X1 50% Seq./50% Ran. 4KB Write 4KB Aligned 180 Minute Screenshot :



# The untold truth about Flash SSD Write IOPS degradation?



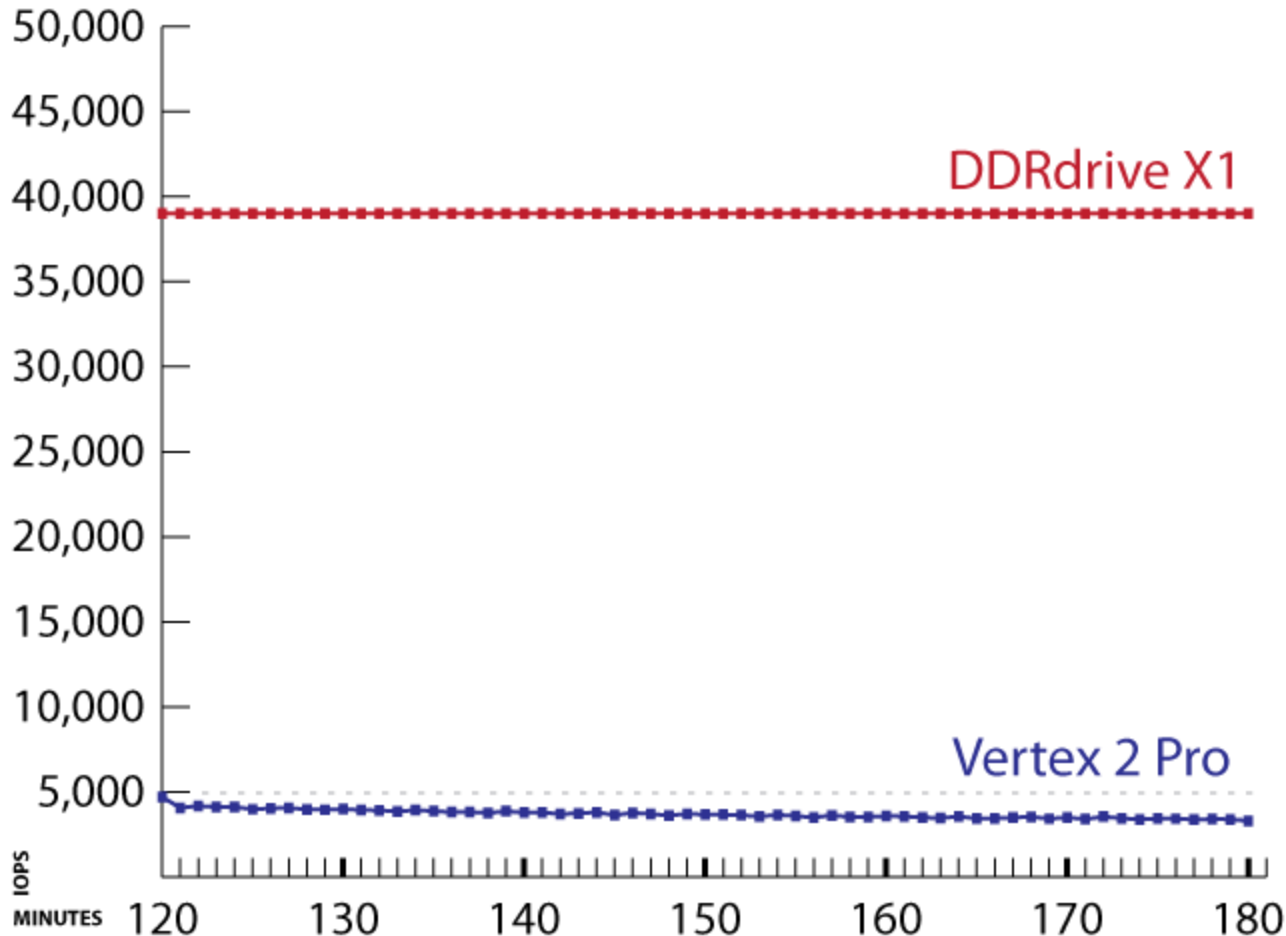
4KB Random Writes

Slide 1 of 2.

**SECURE ERASE**  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

IOmeter 2006.07.27  
IO Alignment = 4KB  
Transfer Size = 4KB  
Transfer Data = Pseudo Random  
100% Random Distribution  
100% Write Distribution  
Queue Depth = 32  
Disk Workers = CPU (2)  
Target Disk = PhysicalDrive  
Time 0 = Start of Test (1 sec)  
Time 1+ = Last Update (60 sec)  
TRIM Not Passed to Devices  
Intel ICH10R Chipset  
Intel RST 9.6.0.1014 Driver  
OCZ Vertex 2 Pro 50GB FW 1.11

# The untold truth about Flash SSD Write IOPS degradation?



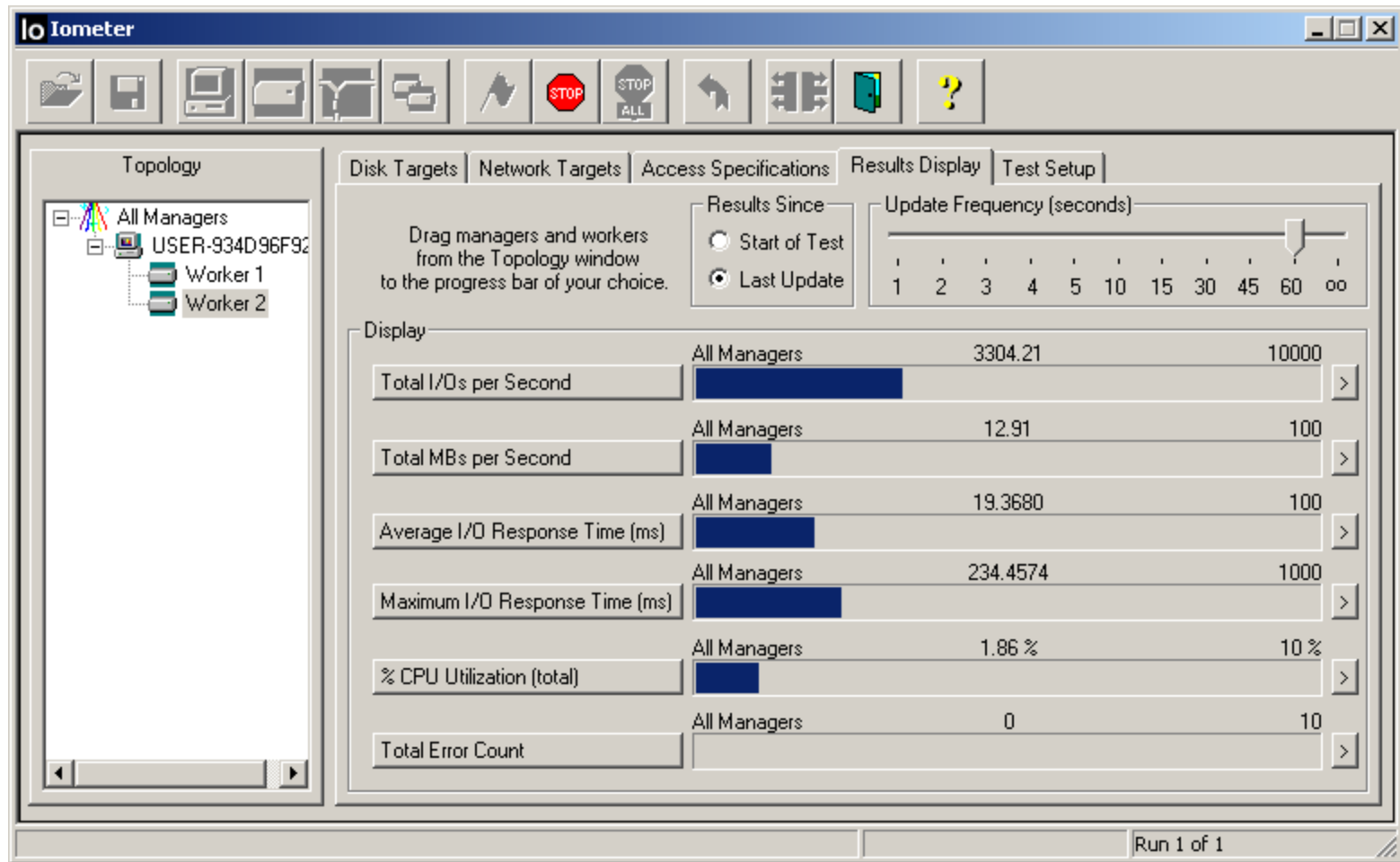
4KB Random Writes

Slide 2 of 2.

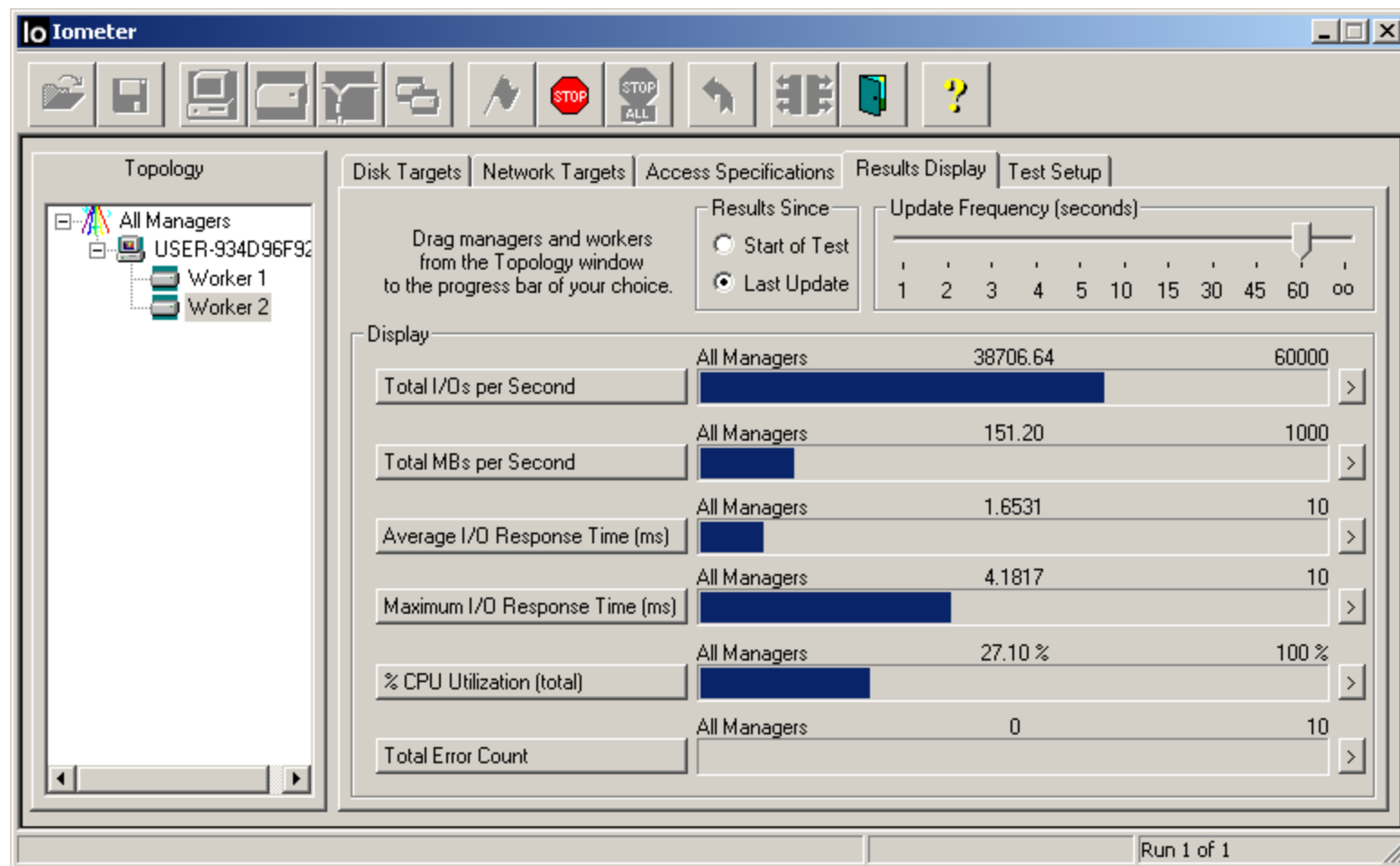
SECURE ERASE  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

IOmeter 2006.07.27  
IO Alignment = 4KB  
Transfer Size = 4KB  
Transfer Data = Pseudo Random  
100% Random Distribution  
100% Write Distribution  
Queue Depth = 32  
Disk Workers = CPU (2)  
Target Disk = PhysicalDrive  
Time 0 = Start of Test (1 sec)  
Time 1+ = Last Update (60 sec)  
TRIM Not Passed to Devices  
Intel ICH10R Chipset  
Intel RST 9.6.0.1014 Driver  
OCZ Vertex 2 Pro 50GB FW 1.11

# Vertex 2 Pro 100% Random 4KB Write 4KB Aligned 180 Minute Screenshot:



# DDRdrive X1 100% Random 4KB Write 4KB Aligned 180 Minute Screenshot:



## Flash SSD Write IOPS Degradation:

With typical ZIL Accelerator use, Flash based SSDs succumb to dramatic write IOPS **degradation** in as short as 10 minutes after device is unpackaged or Secure Erased. As illustrated, this behavior is **not reversed** with device inactivity. Contrast with a DRAM SSD (DDRdrive X1) where performance stays constant not only over the entire product lifetime but with any and all write IOPS workloads (sequential/random/mixed distributions).

*In summary:* The high **sustained** write IOPS usage requirement of the ZIL Accelerator is in direct conflict with the write IOPS degradation of a Flash based SSD.

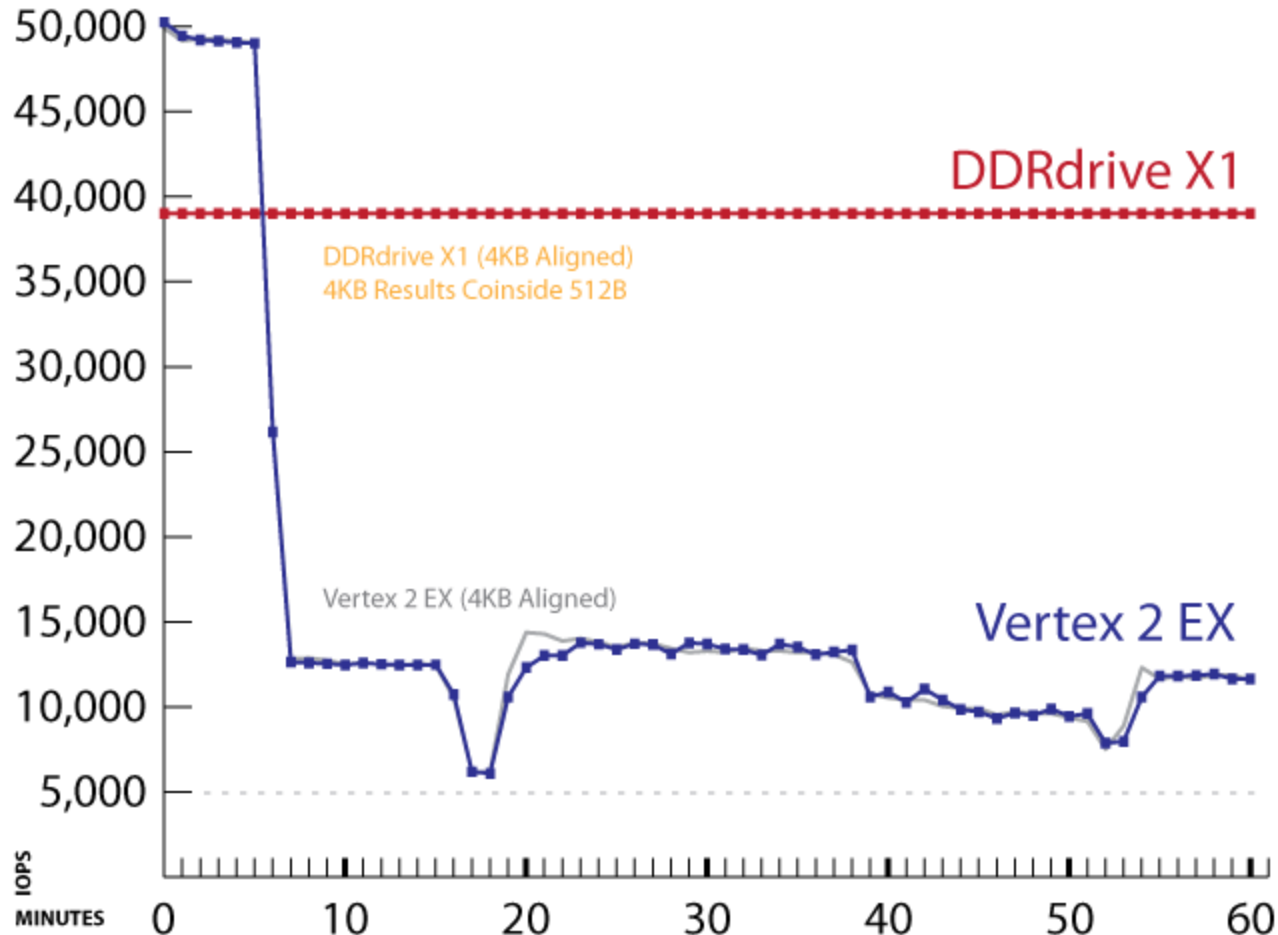
## Questions to be answered:

- What is the ZIL (ZFS Intent Log)?
- Common characteristics of both ZIL Accelerator SSD types?
- Why does the ZIL Accelerator attachment interface matter?
- ZIL Accelerator access pattern random and/or sequential?
- The untold truth about Flash SSD write IOPS degradation?
- **How does IO/partition alignment affect IOPS performance?**
- How do ZIL Accelerator SSD types compare and contrast?

## ZIL Accelerator IO/Partition Alignment?

Both IO and partition alignment versatility are a factor in determining the best suited SSD type for the ZIL Accelerator. A Flash SSD, unlike a DRAM SSD, has alignment requirements for optimum write IOPS and longevity. A DRAM SSD has no such limitations or requirements. The following slides show the same three IO distributions tested prior with both sector (512B) and 4KB alignment write IOPS results superimposed onto a single graph for comparison.

# How does IO/partition alignment affect IOPS performance?



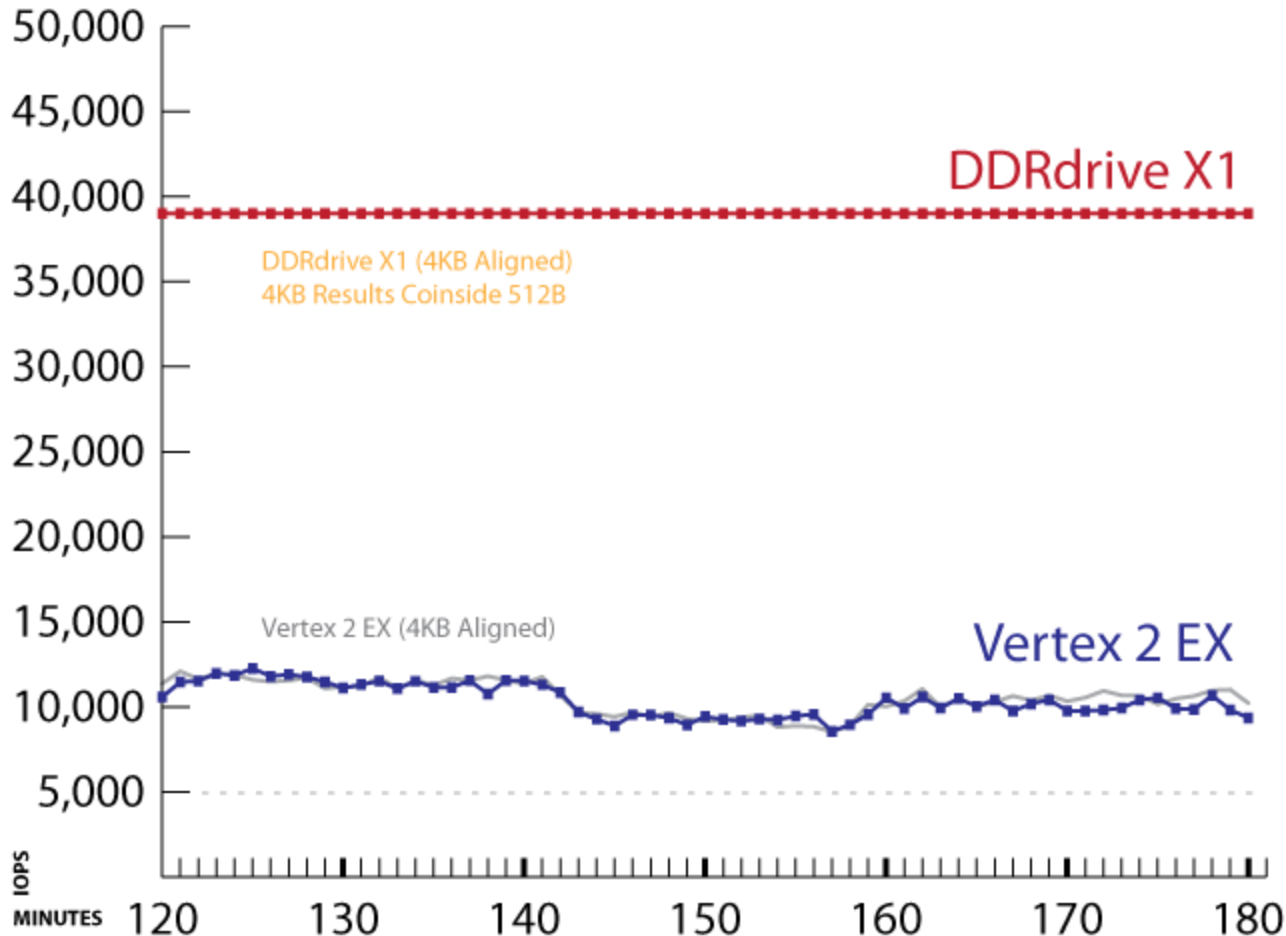
4KB Sequential Writes

Slide 1 of 2.

**SECURE ERASE**  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

lometer 2006.07.27  
**IO Alignment = 512B**  
Transfer Size = 4KB  
Transfer Data = Pseudo Random  
100% Sequential Distribution  
100% Write Distribution  
Queue Depth = 32  
Disk Workers = CPU (2)  
Target Disk = PhysicalDrive  
Time 0 = Start of Test (1 sec)  
Time 1+ = Last Update (60 sec)  
TRIM Not Passed to Devices  
Intel ICH10R Chipset  
Intel RST 9.6.0.1014 Driver  
OCZ Vertex 2 EX 50GB FW 1.11

# How does IO/partition alignment affect IOPS performance?



4KB Sequential Writes

Slide 2 of 2.

SECURE ERASE  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

lometer 2006.07.27

**IO Alignment = 512B**

Transfer Size = 4KB  
Transfer Data = Pseudo Random

100% Sequential Distribution  
100% Write Distribution

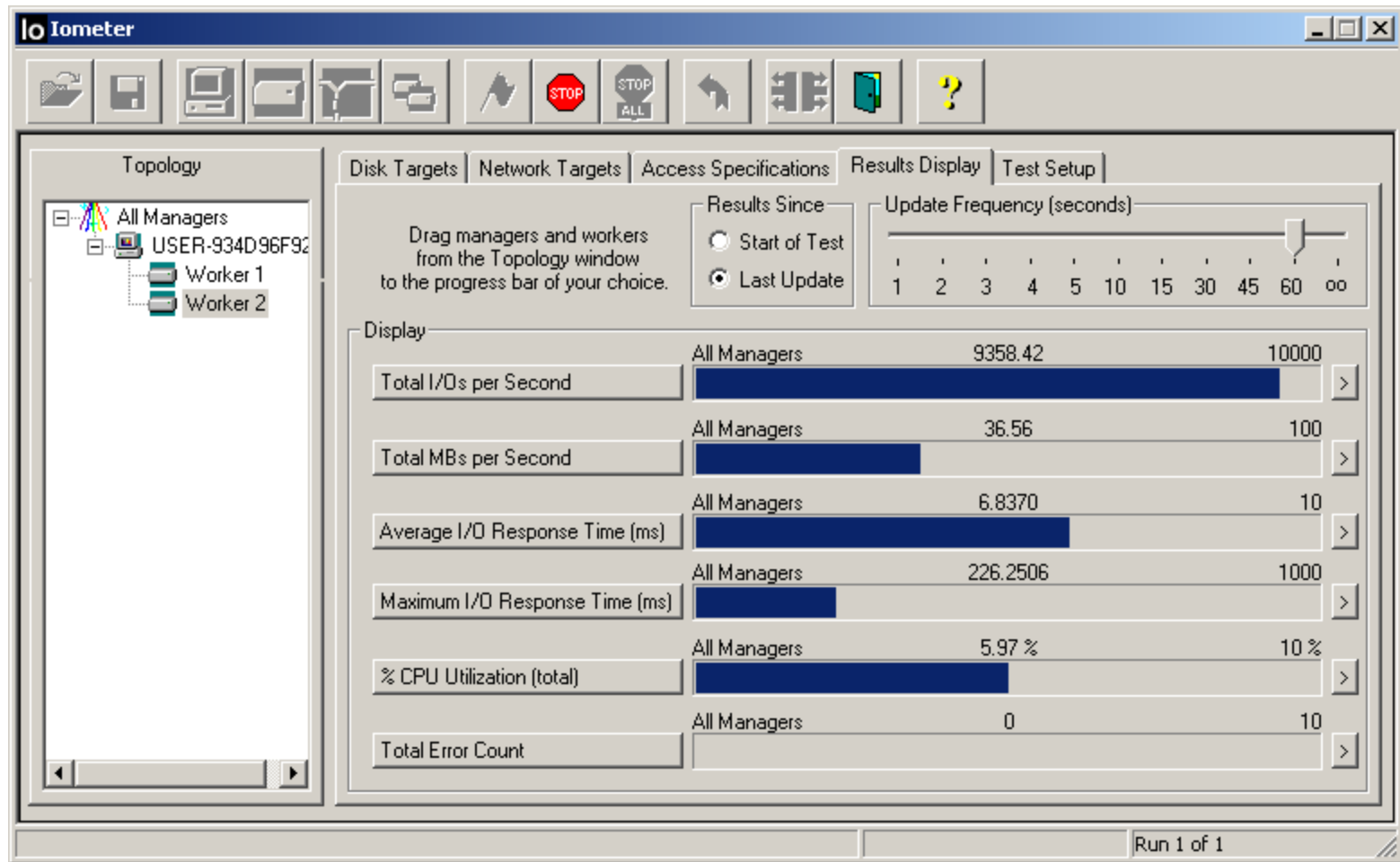
Queue Depth = 32  
Disk Workers = CPU (2)  
Target Disk = PhysicalDrive

Time 0 = Start of Test (1 sec)  
Time 1+ = Last Update (60 sec)

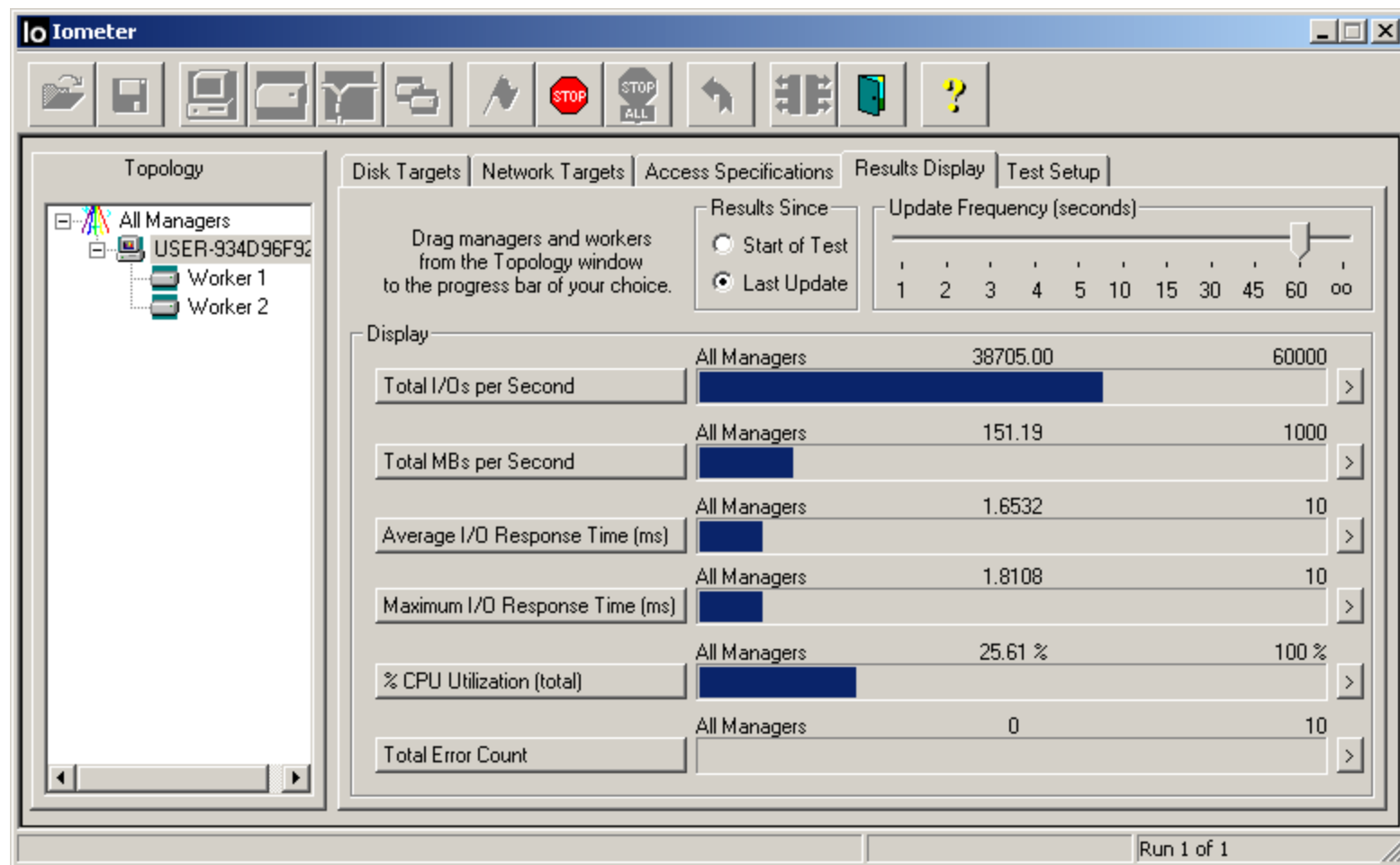
TRIM Not Passed to Devices

Intel ICH10R Chipset  
Intel RST 9.6.0.1014 Driver  
OCZ Vertex 2 EX 50GB FW 1.11

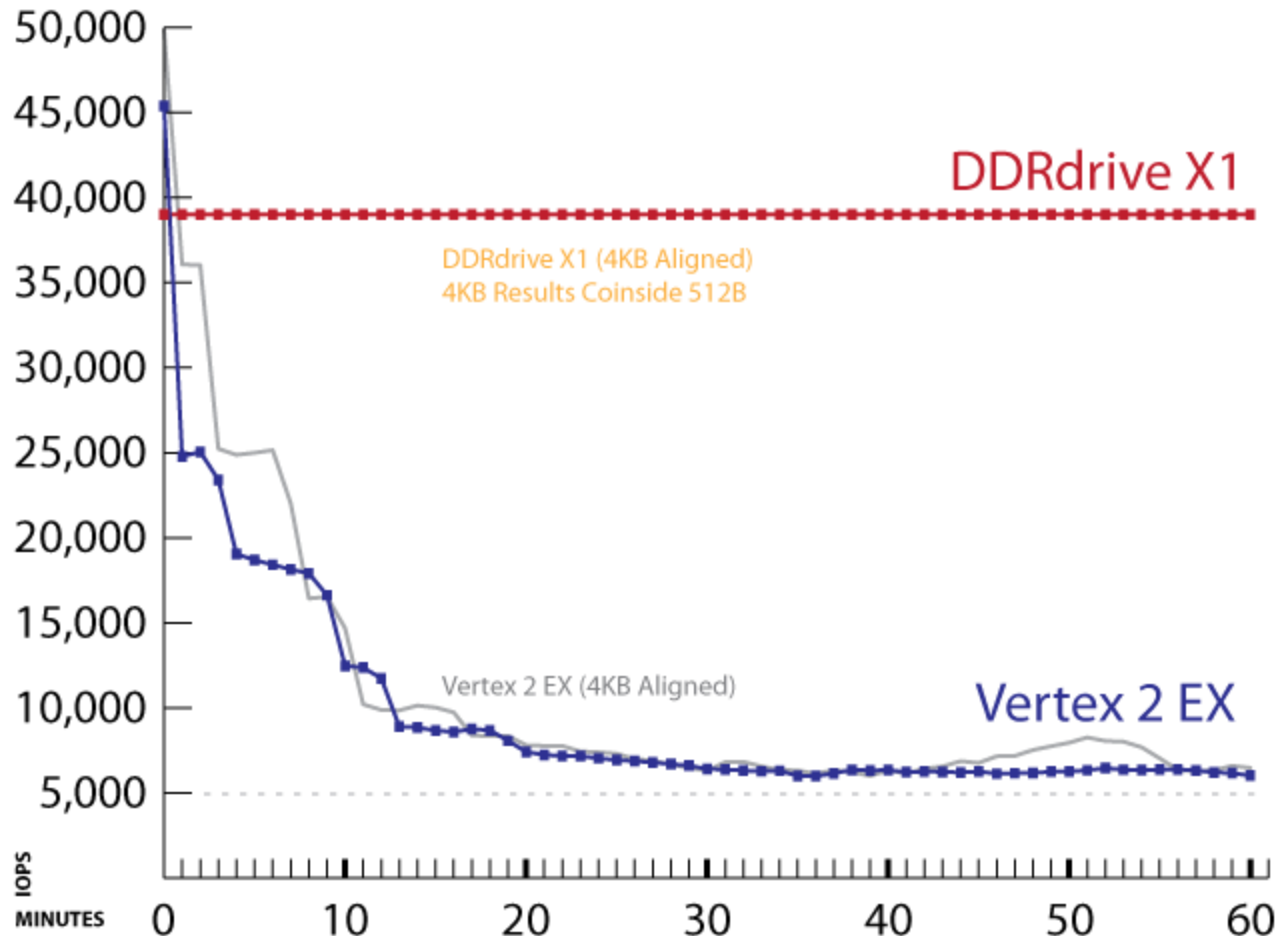
# Vertex 2 EX 100% Sequential 4KB Write 512B Aligned 180 Minute Screenshot:



# DDRdrive X1 100% Sequential 4KB Write 512B Aligned 180 Minute Screenshot:



# How does IO/partition alignment affect IOPS performance?



4KB Mixed Writes

Slide 1 of 2.

**SECURE ERASE**  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

lometer 2006.07.27

**IO Alignment = 512B**

Transfer Size = 4KB  
Transfer Data = Pseudo Random

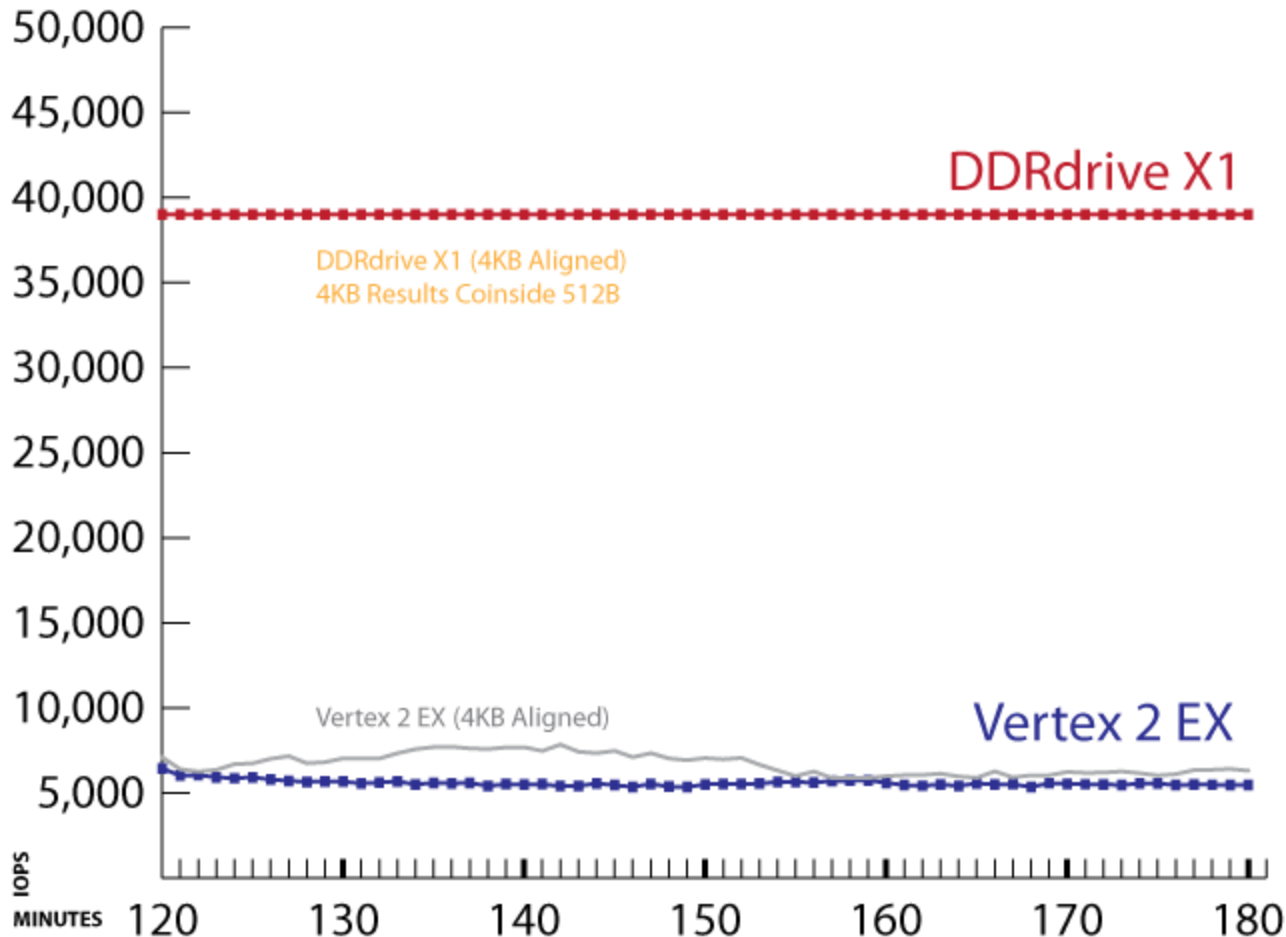
50% Sequential Distribution  
50% Random Distribution  
100% Write Distribution

Queue Depth = 32  
Disk Workers = CPU (2)  
Target Disk = PhysicalDrive

Time 0 = Start of Test (1 sec)  
Time 1+ = Last Update (60 sec)

TRIM Not Passed to Devices  
Intel ICH10R Chipset  
Intel RST 9.6.0.1014 Driver  
OCZ Vertex 2 EX 50GB FW 1.11

# How does IO/partition alignment affect IOPS performance?



4KB Mixed Writes

Slide 2 of 2.

SECURE ERASE  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

lometer 2006.07.27

**IO Alignment = 512B**

Transfer Size = 4KB  
Transfer Data = Pseudo Random

50% Sequential Distribution  
50% Random Distribution  
100% Write Distribution

Queue Depth = 32  
Disk Workers = CPU (2)  
Target Disk = PhysicalDrive

Time 0 = Start of Test (1 sec)  
Time 1+ = Last Update (60 sec)

TRIM Not Passed to Devices  
Intel ICH10R Chipset  
Intel RST 9.6.0.1014 Driver  
OCZ Vertex 2 EX 50GB FW 1.11

# Vertex 2 EX 50% Seq./50% Ran. 4KB Write 512B Aligned 180 Minute Screenshot:

The screenshot shows the Iometer software interface. On the left, the 'Topology' window displays a tree structure with 'All Managers' expanded to show 'USER-934D96F92', 'Worker 1', and 'Worker 2'. The main window has tabs for 'Disk Targets', 'Network Targets', 'Access Specifications', 'Results Display', and 'Test Setup'. The 'Results Display' tab is active, showing a table of performance metrics for 'All Managers'.

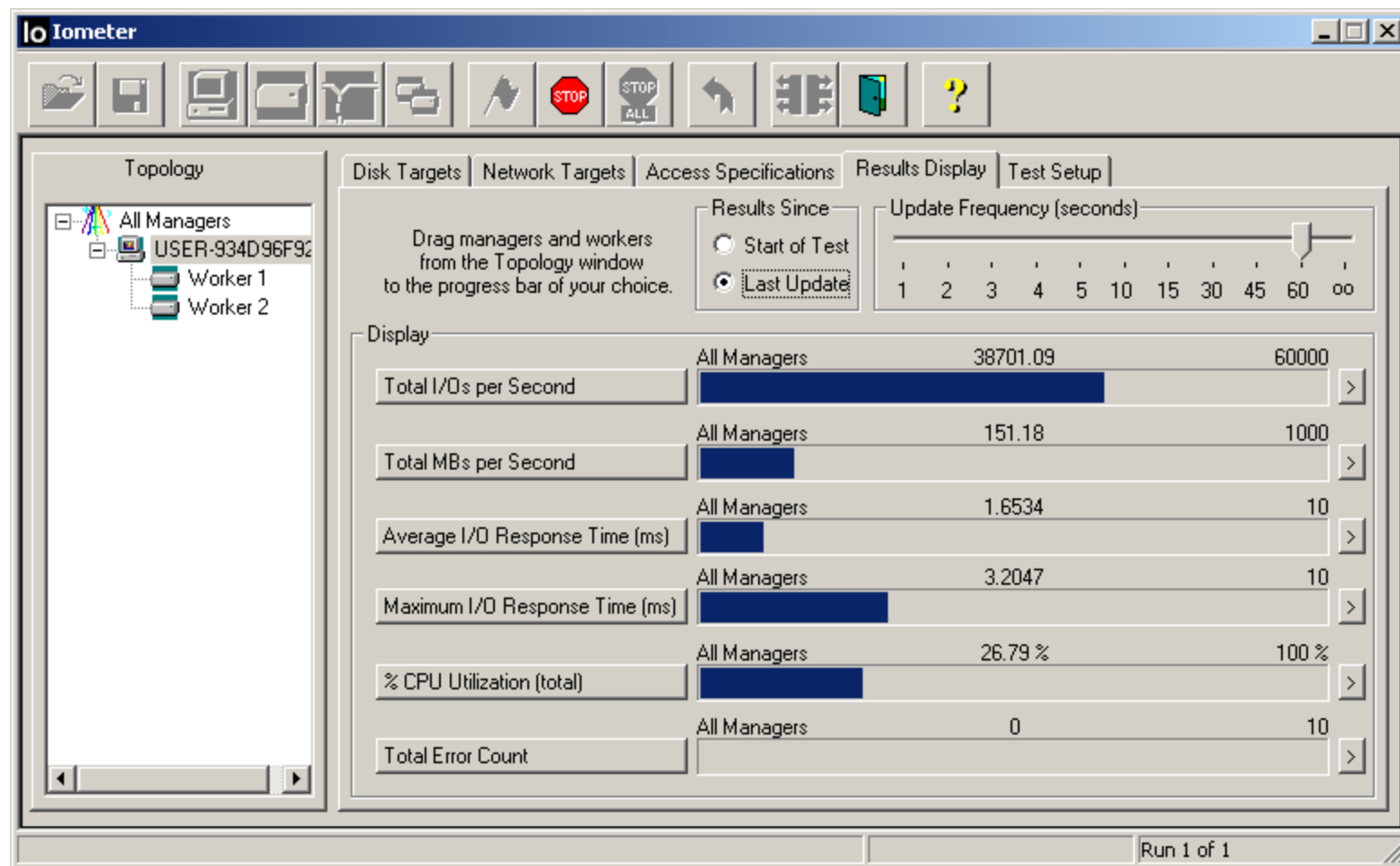
Results Since:  Start of Test  Last Update

Update Frequency (seconds): 1 2 3 4 5 10 15 30 45 60 oo

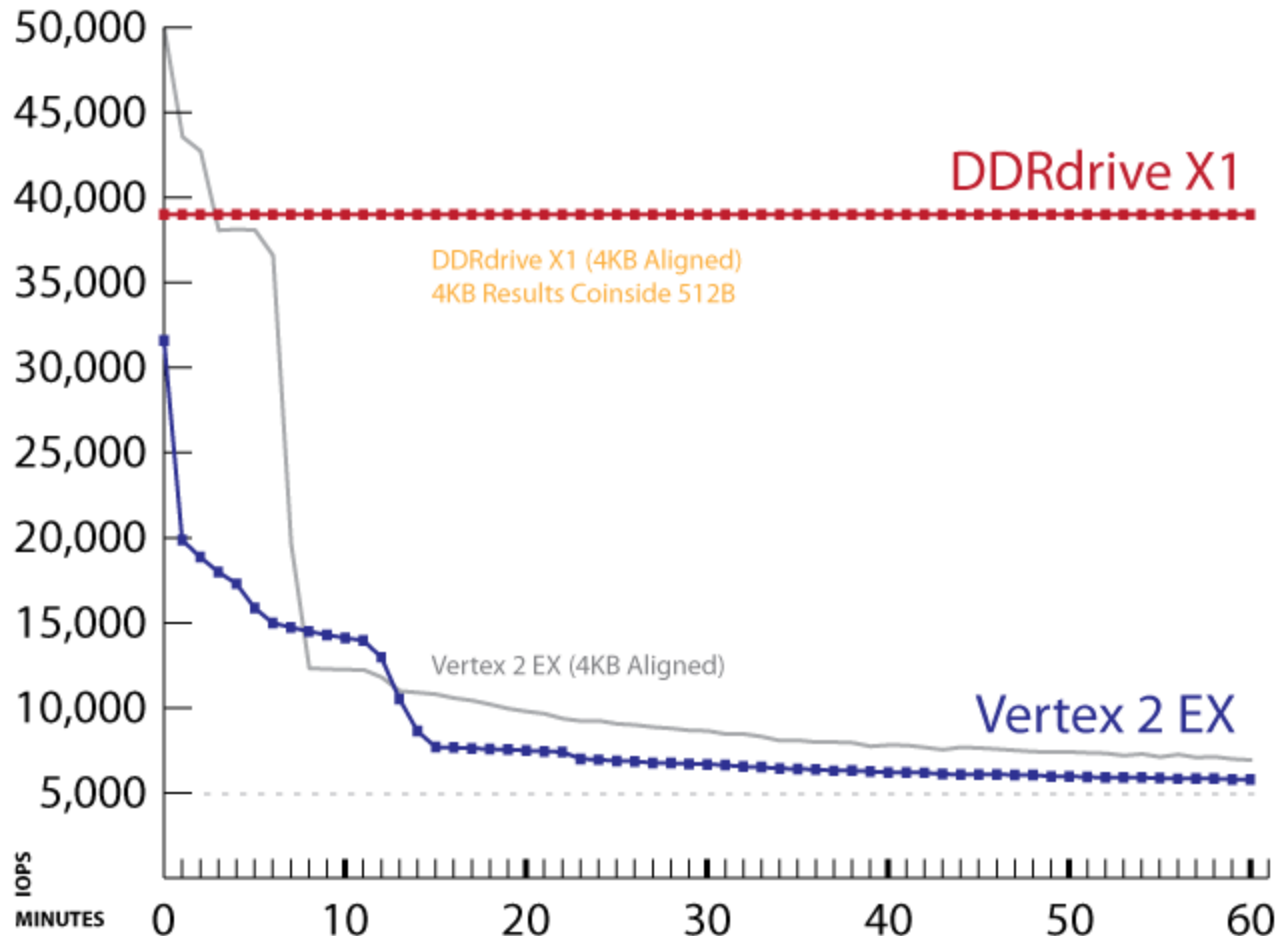
Metric	Value	Scale
Total I/Os per Second	5460.44	10000
Total MBs per Second	21.33	100
Average I/O Response Time (ms)	11.7224	100
Maximum I/O Response Time (ms)	225.5198	1000
% CPU Utilization (total)	3.87 %	10 %
Total Error Count	0	10

Run 1 of 1

# DDRdrive X1 50% Seq./50% Ran. 4KB Write 512B Aligned 180 Minute Screenshot:



# How does IO/partition alignment affect IOPS performance?



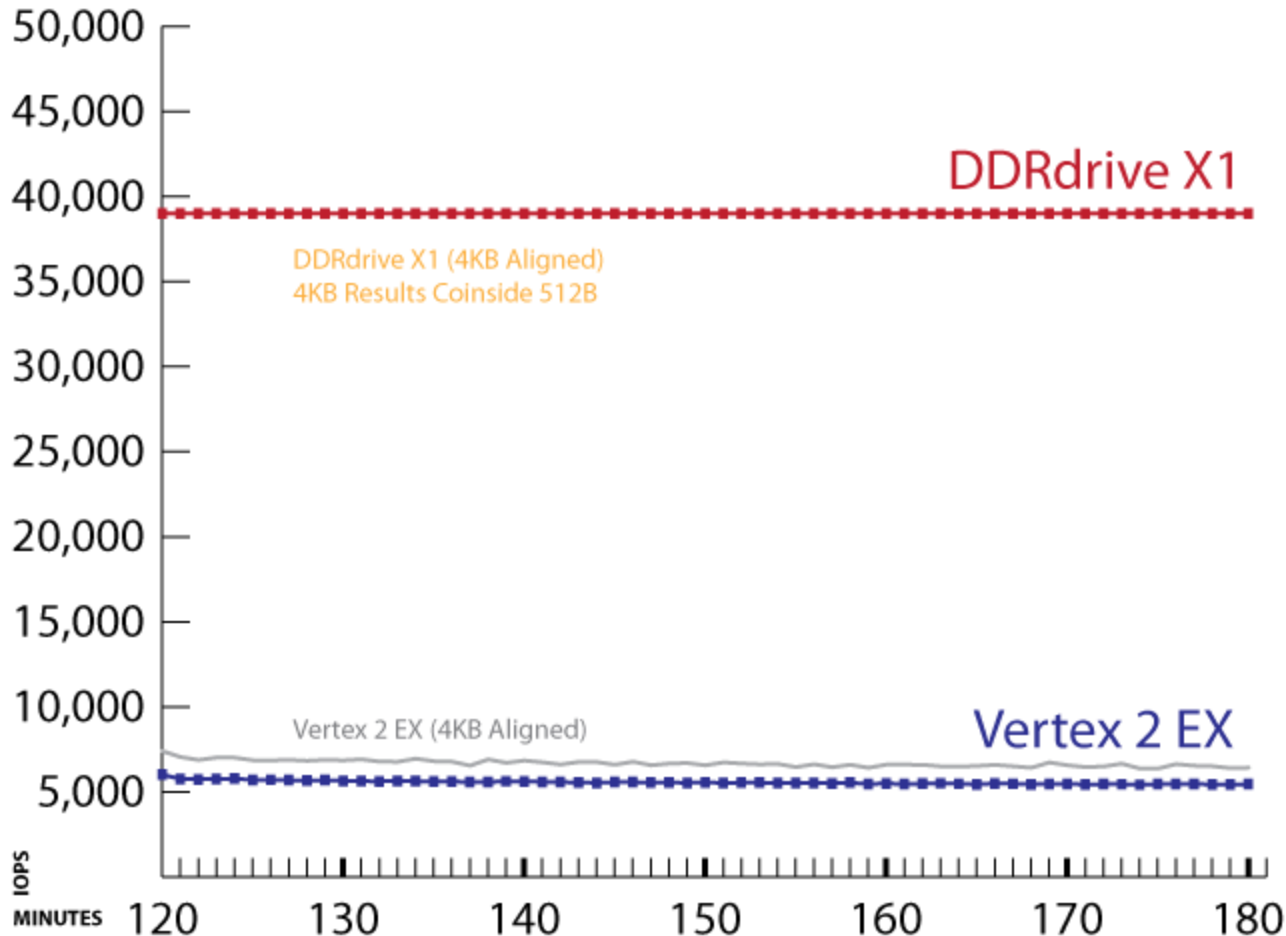
4KB Random Writes

Slide 1 of 2.

**SECURE ERASE**  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

lometer 2006.07.27  
**IO Alignment = 512B**  
Transfer Size = 4KB  
Transfer Data = Pseudo Random  
100% Random Distribution  
100% Write Distribution  
Queue Depth = 32  
Disk Workers = CPU (2)  
Target Disk = PhysicalDrive  
Time 0 = Start of Test (1 sec)  
Time 1+ = Last Update (60 sec)  
TRIM Not Passed to Devices  
Intel ICH10R Chipset  
Intel RST 9.6.0.1014 Driver  
OCZ Vertex 2 EX 50GB FW 1.11

# How does IO/partition alignment affect IOPS performance?



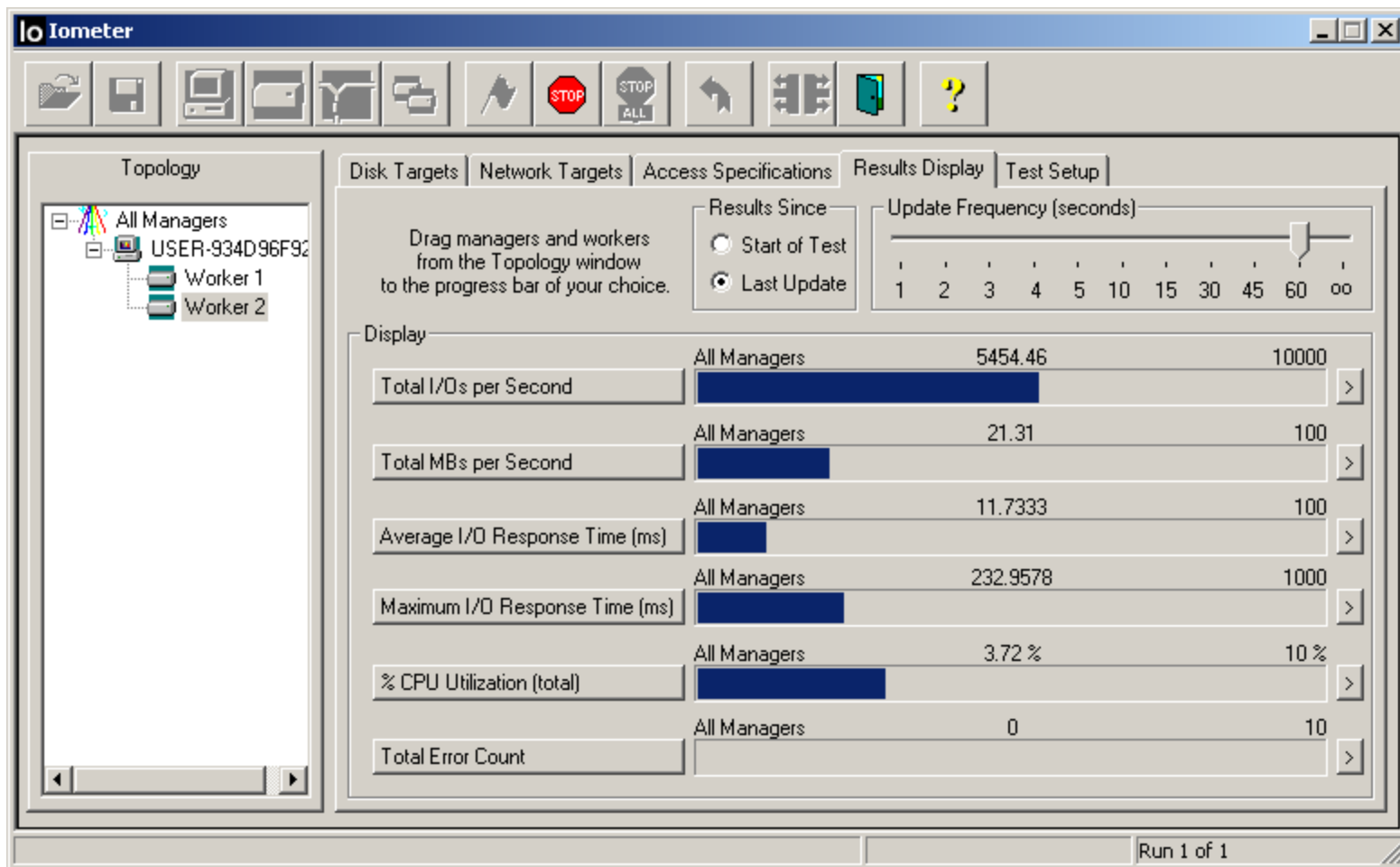
4KB Random Writes

Slide 2 of 2.

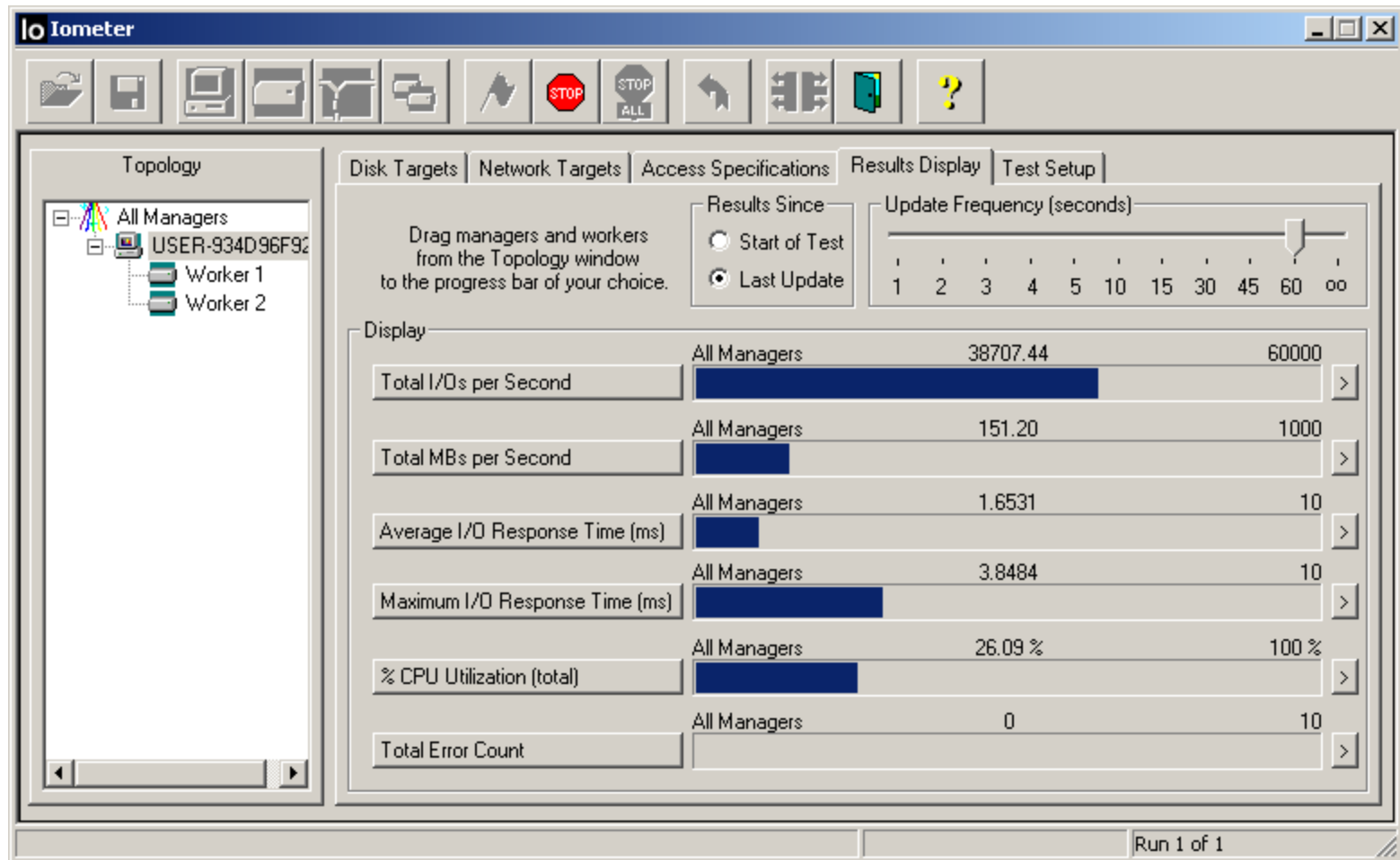
SECURE ERASE  
1HR SUSTAINED  
1HR QUIESCENT  
1HR SUSTAINED

lometer 2006.07.27  
**IO Alignment = 512B**  
Transfer Size = 4KB  
Transfer Data = Pseudo Random  
100% Random Distribution  
100% Write Distribution  
Queue Depth = 32  
Disk Workers = CPU (2)  
Target Disk = PhysicalDrive  
Time 0 = Start of Test (1 sec)  
Time 1+ = Last Update (60 sec)  
TRIM Not Passed to Devices  
Intel ICH10R Chipset  
Intel RST 9.6.0.1014 Driver  
OCZ Vertex 2 EX 50GB FW 1.11

# Vertex 2 EX 100% Random 4KB Write 512B Aligned 180 Minute Screenshot:



# DDRdrive X1 100% Random 4KB Write 512B Aligned 180 Minute Screenshot:



## ZIL Accelerator IO/Partition Alignment:

Flash based SSDs see measurable performance degradation and reduced longevity when either the IO or partition alignment does NOT match internal device boundaries and the IO workload is either random or mixed. A DRAM based SSD (DDRdrive X1) performance does not vary or degrade with **any** IO and/or partition alignment.

*In summary:* A Flash SSD, unlike a DRAM SSD, requires the user to worry about both partition/IO transfer alignment or suffer the consequences (decreased performance and reduced longevity).

## Questions to be answered:

- What is the ZIL (ZFS Intent Log)?
- Common characteristics of both ZIL Accelerator SSD types?
- Why does the ZIL Accelerator attachment interface matter?
- ZIL Accelerator access pattern random and/or sequential?
- The untold truth about Flash SSD write IOPS degradation?
- How does IO/partition alignment affect IOPS performance?
- **How do ZIL Accelerator SSD types compare and contrast?**

# How do ZIL Accelerator SSD types compare and contrast?

<sup>1</sup> FLASH SSD = OCZ Vertex 2 EX / OCZ Vertex 2 Pro <sup>2</sup> DRAM SSD = DDRdrive X1	FLASH SSD <sup>1</sup>	DRAM SSD <sup>2</sup>
Sustained Write IOPS		+
Write IOPS Degradation	×	
Results regardless of IO Data Content.		+
Results regardless of IO Alignment.		+
Results regardless of Partition Alignment.		+
Results regardless of ZFS TRIM Support.		+
Sustained Write IOPS / Dollar (\$)		+
Product Longevity / Durability		+
Product Warranty	3 YEAR	5 YEAR
Requires Separate HBA/Controller	+	
Requires External Power Backup		+

# Thank you!

[www.ddrdrive.com](http://www.ddrdrive.com)