Red Hat Performance Briefs

Removing Performance Bottlenecks in Databases with Red Hat Enterprise Linux and Violin Memory Flash Storage Arrays

Red Hat Performance Engineering

Version 1.0

August 2013

redhat.

violin MEMORY

1801 Varsity Drive™
Raleigh NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park NC 27709 USA

The GPG fingerprint of the security@redhat.com key is:
CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

Send feedback to refarch-feedback@redhat.com

Table of Contents

1. Executive Summary

This paper examines the performance of database applications running on Red Hat Enterprise Linux 6 with Violin Flash Memory arrays.

Violin Flash Memory arrays offer the bandwidth characteristics of a solid state storage device with the connectivity of a SAN. Users can connect to the Memory array using one of the following technologies: Fibre Channel, 10Gb Ethernet, Infiiniband, or PCI-e direct connect.

Violin Memory provided Red Hat with the storage hardware necessary to perform the testing and capture different use cases. As a result of this collaboration, Red Hat found that Violin Flash Memory arrays deliver significant performance gains compared to traditional Fibre channel storage for online transaction processing (OLTP) as well as decision support system (DSS) applications.

2. Test Configuration

2.1 Hardware

Server	4 Sockets, 16 Cores (with Hyper-threading) Four Intel Xeon X7650 CPUs @ 2.26 GHz 128 GB RAM (32 GB per NUMA node)
Fibre Channel Storage	HP – HSV300 (2GB control cache, 2GB Data cache) Capacity: 3.5 TB (28 Physical Disks @ 10,000 RPM)
Violin Memory	Violin 6616 Flash Memory array Formatted for 8 TB of usable space.

Table 1: Hardware Configuration

Violin 6000 series Flash Memory arrays are built from the ground up with multiple controllers to store data on 1000s of individual flash cells in the system. The current-generation single array delivers up to 1 Million IOPS in a consistent, spike-free low-latency manner. This combination of performance and sustained low latency makes the Violin 6000 series Flash Memory array the storage of choice for high IOPS scale out virtual infrastructure configurations, transactional business-critical applications with stringent response time service level agreements, and advanced real-time big data analytics environments.

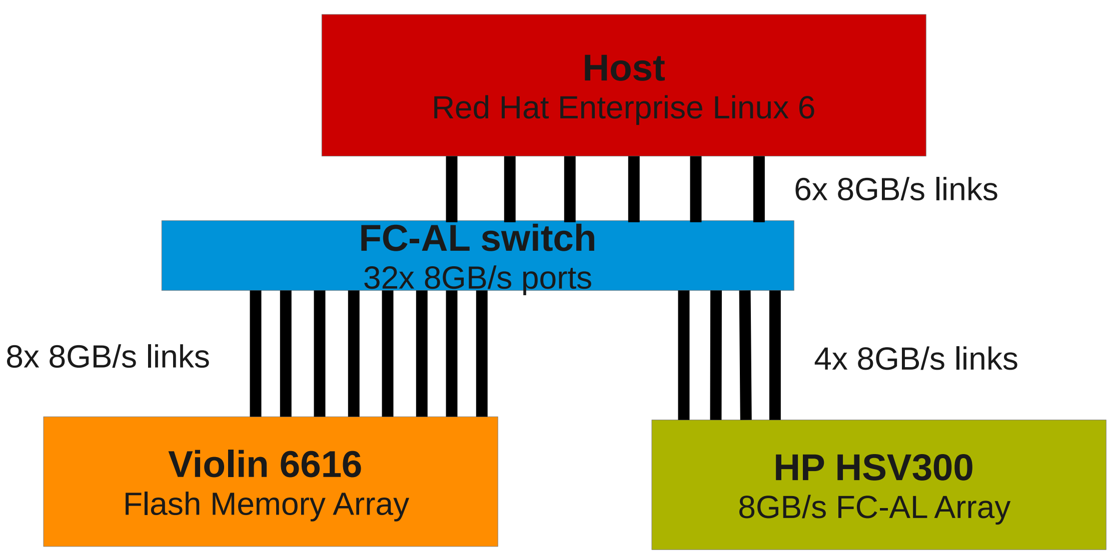
For more information on Violin 6000 series Flash Memory arrays, please see
http://www.violin-memory.com/wp-content/uploads/Violin-Datasheet-6000.pdf

2.2 Software

Operating System	Red Hat Enterprise Linux 6.3 (kernel-2.6.32-330.el6.x86_64)
Violin Memory	G5.2.0
HP – HSV 300	CR1089lep-09000000
Database	Oracle Database 11g Release 2, Enterprise Edition

Table 2: Software Configuration

2.3 Architectural diagram



3. Database Testing

Improvements in application performance or ability to do more work per unit of time have tangible value to their respective owners as every second of machine time counts in modern data centers.

CPU utilization is measured by “work” periods and “idle” periods. As servers have gotten more powerful, most applications tend to underutilize CPU resources. This leads to higher idle periods.

The advent of virtualization and higher-throughput application have put higher demand on CPU utilization, thereby increasing the “work” period. However, much of the “work” is spent on I/O waits. A high IOPS, high-throughput, low-latency storage system, like Violin Flash Memory array, lowers I/O wait time, thereby increasing the efficiency of CPU work periods.

This has the effect of delivering more value out of the existing server investments by delivering more output per unit of time through increased workload capacity or faster computation times.

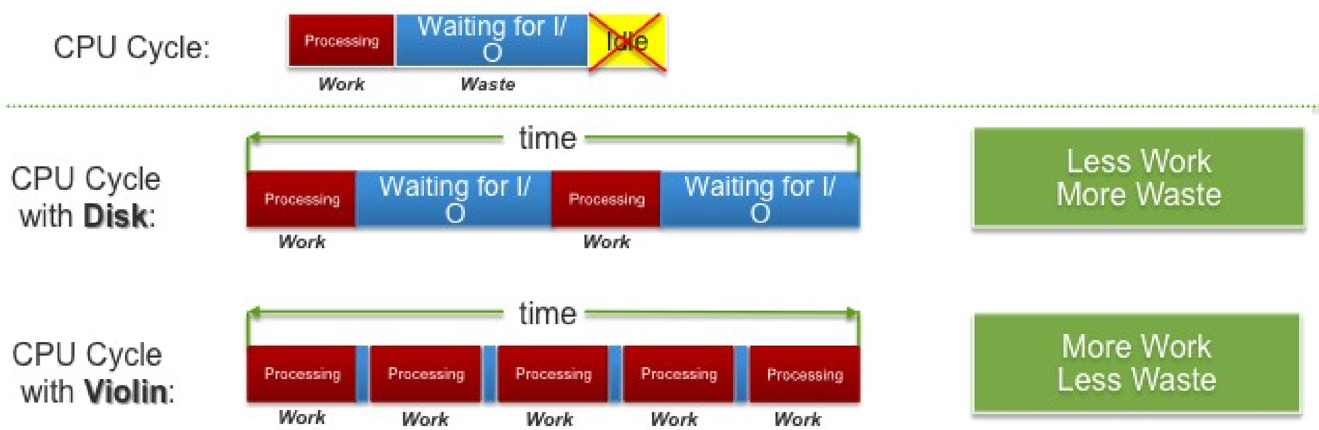


Figure 1: Efficiency gains derived from a high IOPS, high-throughput, low-latency storage system, like Violin Flash Memory array

In the modern enterprise data centers databases could be used for different purposes, but generally they are classified into two broad categories: transaction-processing and business intelligence applications.

To simulate these major database application types, Red Hat Performance Engineering team has conducted testing using Online Transaction Processing (OLTP) and Decision Support System (DSS) workloads. For comparison purposes the baseline tests were conducted using traditional Fibre Channel storage arrays connected to the server via 8GB/s interface. The identical set of tests were then conducted using Violin 6616 Flash Memory array connected to the same server hardware. As demonstrated by results presented below there is a significant advantage of using Violin Flash Memory arrays over the traditional Fibre Channel direct attached storage for both OLTP and DSS applications.

3.1 OLTP Workload

The OLTP workload used in this test was based on the industry standard TPC-C benchmark. The workload was executed with increasing user counts until the server/storage configuration was saturated. Saturation was achieved when the I/O sub-system could not process any more I/O and system statistics exhibited I/O waits. The same database application was then configured and executed on the Violin Flash Memory array and results corresponding to the earlier tests were collected and compared.

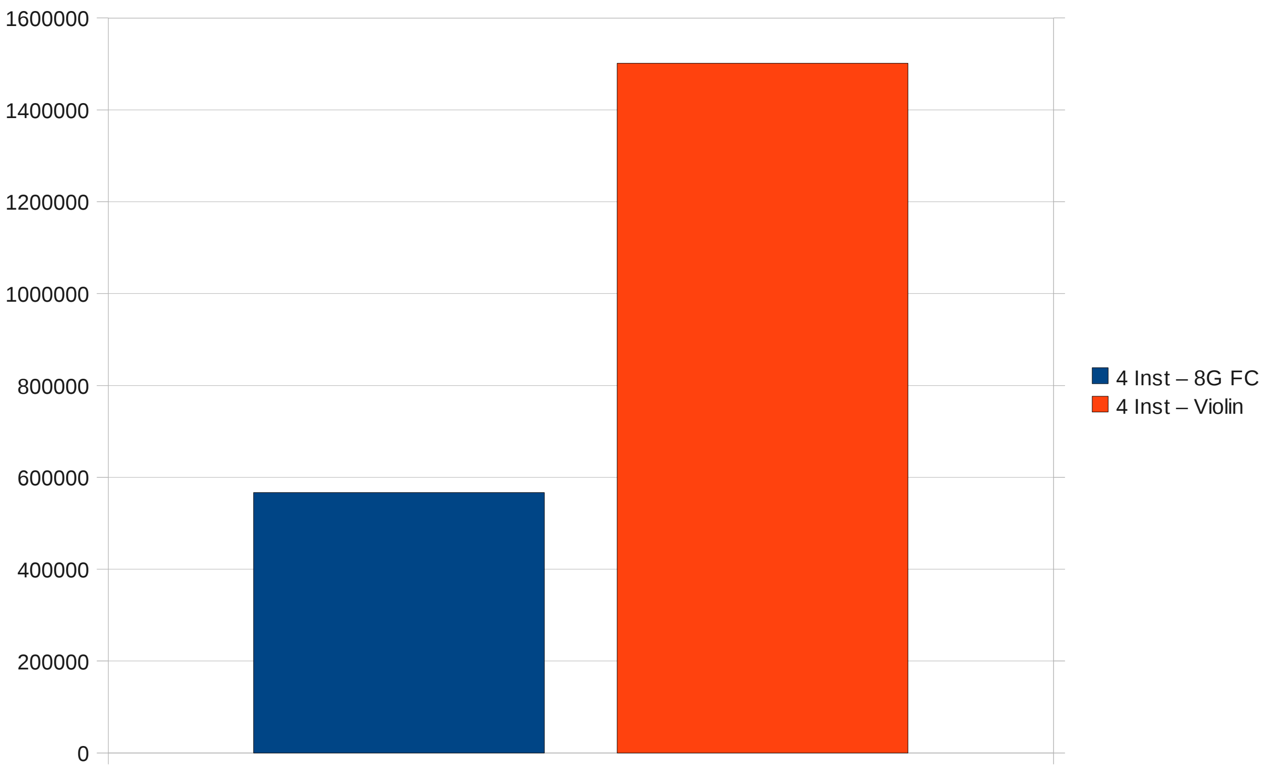


Figure 1: 8Gb/s Fibre Channel array vs. Violin Flash Memory array with OLTP workload (transactions/min)

Figure 1 compares the average transaction rates achieved by two types of storage devices and shows that Violin Flash Memory array performs 2.5 times faster than the traditional Fibre Channel storage array operating at 8GB/s.

All tests were conducted using ext4 file systems with directIO and asynchronous I/O features enabled. DirectIO is a feature of a file system that bypasses the OS file cache thus eliminating any overhead associated with managing that cache. Most databases use shared memory cache and they also support directIO to avoid caching in both the database cache and file cache. Additionally most databases support asynchronous I/O where it could continue to perform read/write operations while outstanding I/O requests are still being completed. This feature significantly improves I/O performance.

Testing in DirectIO mode combines ease of management, usually associated with a file system, and performance characteristics of Linux raw devices.

r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st
9	0	59248	89533560	597836	38359528	0	0	3293	711	2	4	5	1	90	4	0
3	96	59248	86660320	597936	40359396	3	0	208658	1604	14999	16125	4	3	72	20	0
2	120	59248	83222664	598108	43256828	0	0	313328	2947	17511	22504	4	4	58	35	0
4	117	59248	80179328	598228	45913812	0	0	296747	4808	20479	28827	5	4	56	36	0
3	166	59248	76937808	598348	48441568	0	0	280289	8001	24308	35983	6	4	45	44	0
4	168	59248	74123840	598448	50870272	0	0	269492	11524	24517	41876	8	4	41	46	0
5	155	59248	71297712	598576	53314368	0	0	254077	19279	25946	52282	13	5	37	45	0
4	113	59248	68791720	598756	55488300	0	0	221800	29876	28103	64660	18	6	35	41	0
22	177	59248	66534888	598864	57203992	0	0	176918	38772	31972	76900	23	7	25	45	0
8	195	59248	64707472	598992	58785876	0	0	159896	45378	30485	79101	26	7	25	42	0
2	170	59248	62904040	599112	60368824	0	0	151105	52399	30880	81151	29	8	24	39	0
13	187	59248	61185424	599208	61891892	0	0	139602	57218	31397	81210	32	8	24	37	0
36	179	59248	59575004	599332	63285200	0	0	132840	63933	31827	81791	35	8	23	34	0
17	168	59248	58079844	599448	64574840	0	0	126559	67028	31607	79806	37	8	22	33	0
22	156	59248	56440884	599552	65994872	0	0	134184	70396	31849	81092	39	8	19	33	0
2	120	59248	54531980	599668	67695504	0	0	125102	72014	31995	79357	40	9	17	34	0
39	153	59248	52797724	599784	69205728	0	0	131045	76164	32775	81431	43	9	15	33	0
24	144	59248	51163000	599888	70627168	0	0	129921	77348	33596	82463	43	9	14	34	0

Figure 2: vmstat output collected during Fibre Channel array test

Figure 2 captures the *vmstat* output during the Fibre Channel array testing that shows the maximum I/O rate being limited to approximately 75 MB/s (as seen in the “bo” column). As the load increases and the system limit is reached, processor utilization starts to go up as well. However, majority of CPU time is being spend waiting for I/O (as seen under the “wa” column). Consequently, there is a lot of CPU idle time (as seen in the “id” column) because processors are waiting for the completion of I/O operations.

Next, let’s compare the *vmstat* data captured during Violin Flash Memory array testing.

r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st
9	0	33012	122465720	317524	7421752	0	0	4110	670	0	0	5	1	89	5	0
2	153	33012	98678400	317664	27025000	0	0	2132801	30022	110355	118759	34	25	18	23	0
150	44	33012	83850960	317784	39967984	0	0	1310195	101060	89839	109747	70	25	0	5	0
111	25	33012	77896032	317908	45265580	0	0	452055	132845	57645	100892	80	19	0	1	0
139	17	33012	74218384	318016	48539952	0	0	245796	140245	51181	101968	82	17	1	1	0
146	7	33012	71262640	318132	51195476	0	0	170008	143407	47856	102488	83	16	1	1	0
169	13	33012	68700800	318260	53478072	0	0	132342	146024	45869	102531	83	15	1	0	0
103	11	33012	66368636	318360	55547764	0	0	109217	147246	44374	102589	84	15	1	0	0
81	14	33012	64125764	318568	57534604	0	0	97286	178635	47588	104168	84	15	1	1	0
129	11	33012	62005436	318696	59430372	0	0	83317	147141	48753	104353	84	14	1	1	0
125	10	33012	60039908	318812	61179712	0	0	74282	148675	48468	105153	84	14	1	0	0
116	12	33012	58042980	318944	62956152	0	0	69971	152767	48039	104785	84	14	1	0	0
136	16	33012	56143800	319072	64668444	0	0	63573	149157	47423	104987	85	14	1	0	0
108	7	33012	54285624	319192	66316460	0	0	58562	149769	47307	104767	85	14	1	0	0

Figure 3: vmstat output collected during Violin Flash Memory array test

As seen in Figure 3, the Violin Flash Memory array is able to manage the I/O requirements of the workload much better.

Comparing same columns from both charts, there are no I/O waits (under the “wa” column) and very little idle time (under the “id” column). This indicates that the system is not constrained by I/O waits and is operating at its full potential. By using Violin Flash Memory arrays for this workload, we were able to eliminate the I/O bottleneck, run over 100 processes simultaneously (as seen in “r” column) and improve the overall system performance by 2.5 times.

3.2 DSS Workload

This workload is similar to the industry-standard TPC-H benchmark and consisted of 22 parallel queries.

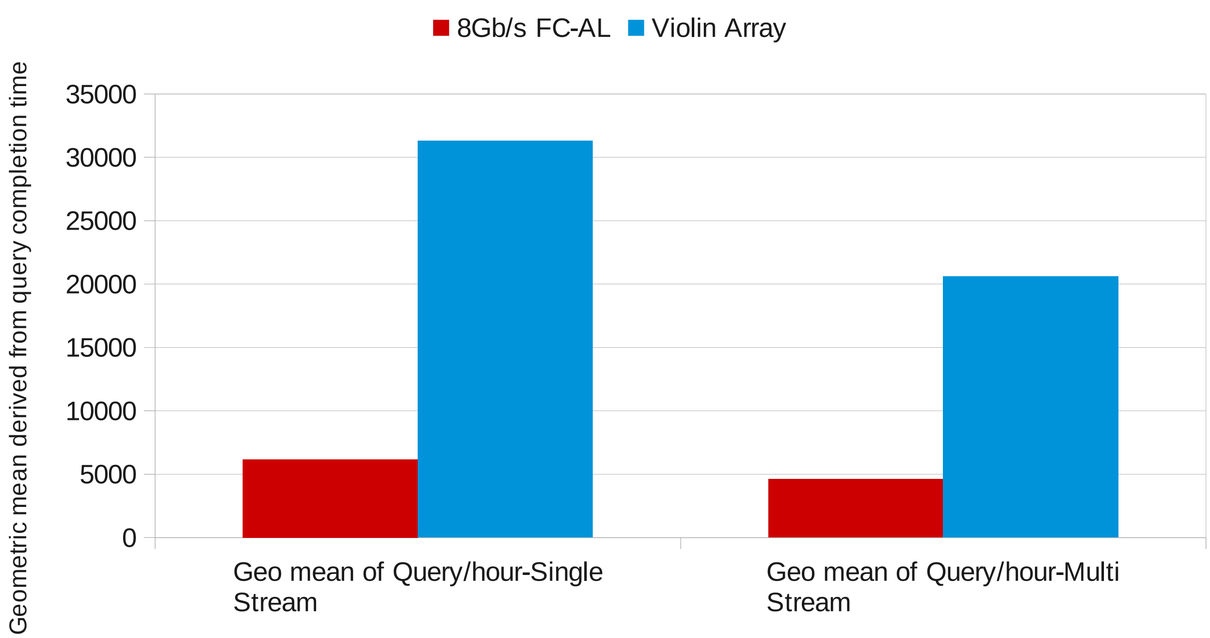


Figure 4. 8Gb/s Fibre Channel array vs Violin Flash Memory array with DSS workload

To effectively compare database workloads running on Violin Flash Memory array and 8GB/s Fibre Channel arrays, the Geometric mean of the Query/Hour metric was selected. Also note that during multi-stream testing five simultaneous streams were executed.

As evidenced by Figure 4, measurements demonstrate five times better throughput when using Violin Flash Memory array compared to 8GB/s Fibre Channel array. Since this workload has a tendency to stress both CPU and I/O operations on the system during the most rigorous testing using multiple streams, the Violin Flash Memory array demonstrated 4.5 times better performance than 8GB/s Fibre channel array.

4. Conclusion

Data collected during both OLTP and DSS tests shows that Violin Flash Memory arrays have high throughput and low latency characteristics that can significantly improve performance of different production workloads. The superior random I/O capabilities of the Violin Flash Memory arrays offer a dramatic advantage over traditional Fibre Channel storage.

One of the advantages of Violin Flash Memory arrays is that they can be accessed by multiple hosts simultaneously. This feature simplifies data access and enables application migrations in case of unexpected disaster and could be very effective in high availability solutions such as those using Red Hat GFS2 Clusters and Red Hat Enterprise Virtualization.

In many cases, the Violin Flash Memory arrays can simply replace the back end storage using either standard Fibre channel protocol (as tested here) or network interfaces. This allows users to deploy Violin Flash Memory arrays under Red Hat Enterprise Linux without any changes to their IT infrastructure or software applications. Additionally, the Violin Flash Memory arrays are easy to configure and manage with user-friendly interfaces making it a compelling enhancement to either the small-scale or mission-critical database deployments.

Another implicit benefit of using Violin Flash Memory arrays under Red Hat Enterprise Linux is that no proprietary or additional device drivers are required for deployments. Whether Fibre Channel, 10GbE or Infiniband adapters are used in the system to enable external connectivity, the Violin Flash Memory array typically utilizes those existing connections from the host and leverages the device drivers that are already included with the Red Hat Enterprise Linux operating system. That greatly simplifies the out-of-the-box deployments of Violin Flash Memory arrays under Red Hat Enterprise Linux and eliminates the need to certify another piece of infrastructure.