EMC VFCache™ Flash Caching Solution Evaluation

Evaluation report prepared under contract with EMC Corporation

Introduction

IT professionals are seeking ways to increase the performance of mission-critical applications and are looking to solid state storage as a way to make this happen. However, it is not economically feasible to replace large quantities of spinning hard disk storage with solid state storage. Are there ways to achieve significant performance gains with a small amount of solid state storage that are simple to manage?

SSD caching is becoming an excellent way to deploy solid state storage. SSD caching is able to significantly improve performance because it puts a copy of frequently accessed data into the cache, benefiting any application whose data is in the cache. This causes significant performance gains for various application workloads, especially database workloads, without having to store all the application data on solid state storage.

EMC commissioned Demartek to evaluate its VFCache Flash caching solution for its ability to improve several types of database workloads running on EMC Symmetrix VMAX and EMC VNX storage systems.

Evaluation Summary

We were able to observe 2.5 times to 3.5 times performance gains with Oracle and Microsoft SQL Server OLTP database workloads by deploying EMC VFCache on the database server. EMC VFCache also outperformed a competitive solution from Fusion-io in our tests. EMC VFCache is transparent to the applications and provides significant performance gains with very simple management.

We believe that EMC VFCache is a very cost-effective way to improve the performance of heavy-duty I/O workload applications.

Solid State Storage Technology

Solid state storage (SSS) isn't only transforming the storage industry; it's making waves across the entire computing industry. It has moved from a consumer-grade technology used in cameras and "thumb drives" to heavy duty workloads in enterprise datacenters, providing high-performance persistent storage for critical applications.

When compared to hard disk drives, solid state storage can perform significantly faster. Comparing a single PCIe solid state storage device to a single hard disk drive (HDD), the performance can be hundreds of times faster. PCIe solid state storage devices in particular have very low latency because they operate directly on the system bus of the host server. The best enterprise hard drives have an average of approximately 2ms of seek time latency for every request, and not every storage system enables the cache memory on the drives because of data protection concerns. So even if the SSD technology in use had the exact same performance as a hard drive, the SSD technology would provide better overall latency because it has no seek time. Imagine running a large batch of complex database transactions where every I/O is subject to the seek time latency of good enterprise hard drives. Then imagine that same batch of complex database operations without the seek time latency and with faster storage devices. You will see why SSDs are so good for database applications and other enterprise workloads.

Solid state storage devices are computer storage devices that use memory technology for the storage media rather than traditional magnetic media such as hard disk drives or tape drives. These SSS devices have no moving parts and can be made with DRAM technology or Flash memory technology, or sometimes a combination of both. These devices appear to the host operating system as storage devices.

The solid state storage technology gaining the most attention in the enterprise is NAND Flash. NAND Flash is available in two basic types: single-level Cell (SLC) and multi-level cell (MLC). SLC Flash has one bit per cell, and is generally designed for enterprise applications. MLC Flash has more than one bit per cell, and is generally designed for consumer applications. There is a relatively new category known as eMLC (Enterprise MLC) that is MLC Flash but with some of the characteristics of SLC Flash. The enterprise characteristics of eMLC Flash are provided primarily by the intelligence within the low-level flash controllers on the device, and are lower cost than SLC Flash but suitable for enterprise applications. SLC and MLC Flash also differ in their endurance, or number of write cycles available, for each. SLC Flash generally has 10-20 times the write endurance of consumer-grade MLC Flash, and is much more suitable for heavy-duty enterprise applications that require 24x7 operations.

Database administrators, system administrators and application owners have become aware of solid-state storage and the benefits that it brings. They recognize the performance and power consumption benefits, and many are now evaluating various types of solid state storage technologies. Solid state storage is available in a variety of form factors, including the disk drive form factor in a server or storage array and PCIe adapter cards in servers.

One of the primary questions remaining is how best to use solid state storage, and what types of usage provide the best return on investment.

SSD Caching

One approach to using solid state storage is to use it as a cache. Caching for SSDs is determined by host software or intelligence in the storage controller, but the main feature of a cache is that it places a ***copy*** of "hot" data into the SSD cache, while retaining the data in its original storage location that is known to the users and applications. Data is placed into the cache when the data is frequently accessed ("hot") and removed from the cache when the data is no longer frequently accessed ("cold") and other data becomes hot. Because SSD caches operate below the application levels, applications do not need to be modified to take advantage of the SSD cache.

Caching is relatively simple to manage because nearly all the decisions are made by the caching software or controller. Caching benefits any, and potentially every, application whose data is considered "hot" within the scope of data accessible to the cache. SSD caching solutions may differ with respect to particular caching algorithms used, but each one observes the data access patterns and automatically determines what data to place in the cache and when to place it in the cache. The goal of a caching system is to keep as much of the hot data as possible in the cache, boosting performance because the SSDs generally have higher performance and lower latency than spinning hard disk drives. SSD caching, in effect, masks the slower seek performance of HDDs. Some caching solutions will not only cache the hot data but may also pre-fetch data that the caching software believes might also benefit from being in the SSD cache based on the I/O patterns observed. Some caching systems cache only the read operations, while others cache both reads and writes. When writes are cached in an SSD caching system, the usual protections must be in place to maintain data protection similar to the way this data is protected in a storage system DRAM cache.

Because of the high I/O rates that occur in enterprise environments, SLC Flash technology is the best choice for SSD caching applications.

The performance boost provided by SSD caching systems increases over time as the cache fills up. This phenomenon is known as the cache "warm-up" or "ramp-up." This cache warm-up can occur over minutes or hours depending on the I/O access patterns and the size of the cache.

Many SSD caching systems allow the administrator to specify the scope of the cache management by excluding certain volumes or filesystems. Some applications may have unusual data access patterns that might not make them good candidates for the SSD cache.

The other parameter that can be adjusted by the administrator is the amount of SSD cache to use. Some caching systems allow a fixed cache to be split into an amount of SSD to be used for caching purposes and the remaining amount to be used for direct or primary data storage. If the SSD cache is composed of SSD technology that can be increased, such as disk drive form factor, then the administrator may increase the amount of SSD cache simply by adding SSDs.

Read Cache Effect on Writes

When reads are cached, this has the effect on reducing the overall load on the back-end storage system, allowing the storage system more resources to dedicate to other operations such as writes or internal storage system management.

EMC Caching Technologies

EMC provides several caching technologies that can be applied where needed. Data can be cached in the server with EMC VFCache or in external EMC storage systems with DRAM cache and SSD cache.

Caching in the server with VFCache benefits from the intelligence provided by the VFCache card and software drivers. Data access gets a significant performance boost because the VFCache card uses SLC NAND Flash and operates directly on the PCIe bus and provides very low latency and high throughput. This server cache is transparent to the applications running on the host server and these applications do not need to be modified in any way to take advantage of VFCache. VFCache provides a tremendous benefit to I/O operations because once cached, I/O accesses are satisfied within the server generally within microseconds and do not need to travel through an external switch or external interface to get to external storage where accesses are typically measured in milliseconds.

EMC FAST provides SSD performance benefits in the VMAX and VNX external storage systems by providing large amounts of SSD for cache or storage tier in the disk drive form factor. Although this cache or storage tier is not as fast as VFCache in the server, FAST cache with Flash drives in the storage system is larger and can provide a strong performance boost within the storage system. FAST in the VMAX or VNX storage systems can be combined with VFCache in the server to provide an outstanding performance benefit.

EMC VFCache Architecture

EMC's VFCache uses a PCIe solid state storage card in a Windows, Linux or VMware server to cache I/O operations to back-end storage arrays such as the VMAX or VNX families of storage systems. This type of write-through cache provides a dramatic improvement in application response time and accelerates read operations. This cache is transparent to applications and requires no modifications to applications for their I/O operations to be accelerated.

EMC has performed interoperability testing with Fibre Channel connections and with their own storage arrays. However, adding support for other arrays and connection protocols is just a testing effort and it is expected that official statements of support will be made soon.

VFCache is assigned LUNs to which it should apply its cache resources. This can be some or all of the LUNs accessible by that host server. For example, consider a database application that has spread its database files across multiple LUNs and its logs across different LUNs. The LUNs holding the database files would make excellent candidates for VFCache, while the log LUNs for that same database application would not be good candidates for VFCache. The LUNs assigned to VFCache are said to be "accelerated LUNs." EMC VFCache accelerates I/O operations while maintaining full data protection and durability by passing the writes through to the back-end storage array.

The following three examples describe how VFCache currently operates.

Read Hit Example

When an application issues a read request for data from an "accelerated LUN," the VFCache driver determines if this data is already in the cache and can be satisfied by the cache. If this read request can be satisfied by the cache, the requested data is immediately returned to the application without contacting the external storage, resulting in very fast response time.

Read Miss Example

When an application issues a read request for data from an "accelerated LUN," the VFCache driver determines if this data is already in the cache and can be satisfied by the cache. If the requested data is not in the cache, the read request is passed through to the external storage. When the requested data is satisfied by the external storage it is returned to the application in the normal manner. A copy of the requested data is also placed into the cache, if the data is less than or equal to 64KB in size.

Write Example

When an application issues a write request for data to an "accelerated LUN," the VFCache driver passes the write to the external storage in the normal manner. When the write has been completed and acknowledged by the external storage, the application is notified also in the normal manner. The data is also written into the cache, if the data is less than or equal to 64KB in size.

VFCache Performance Results

As a result of this design, a large majority of reads are serviced by VFCache, dramatically improving application response time and overall performance. Typical read latencies for cache hits would be

less than 100 microseconds (< 100µs). Writes are passed through to the storage array in the normal manner and are fully protected by the storage array's normal protection mechanisms.

Because of the 64KB feature, large streaming data or backup operations are not cached, which prevents wasting cache resources for data that is probably going to be accessed infrequently and does not fall under the category of "hot" data.

VFCache and Storage Subsystem Features

As the latest copy of the data is always maintained on the storage array, users can continue to use their existing disaster recovery (DR), high-availability (HA) and other features of the storage array after deploying VFCache.

VFCache Workload Guidance Best Practices

The following are some best practices to use in order to take maximum advantage of VFCache's capabilities:

- **Read Intensive Workloads** – Applications that have a read-heavy workload, where the ratio of reads to writes is at least 50% are good candidates for caching.
- **Small I/O Size** – Applications whose read requests are 64KB or smaller are eligible to be cached with VFCache.
- **Random I/O** – Random I/O activity makes a better candidate for caching rather than sequential I/O. Sequential I/O activity does not often result in "hot" data or repeated access to the same data.
- **Highly Concurrent** – VFCache excels when there are many I/O streams operating in parallel, either from the same application or multiple applications.

Other things to consider are the working set size of the application and the "locality of reference" or data access patterns. If an application fits the guidelines listed above, but accesses its data approximately evenly across all of its data, it may not see as much cache benefit as an application that has "hot spots" of activity.

VFCache Split Card Mode

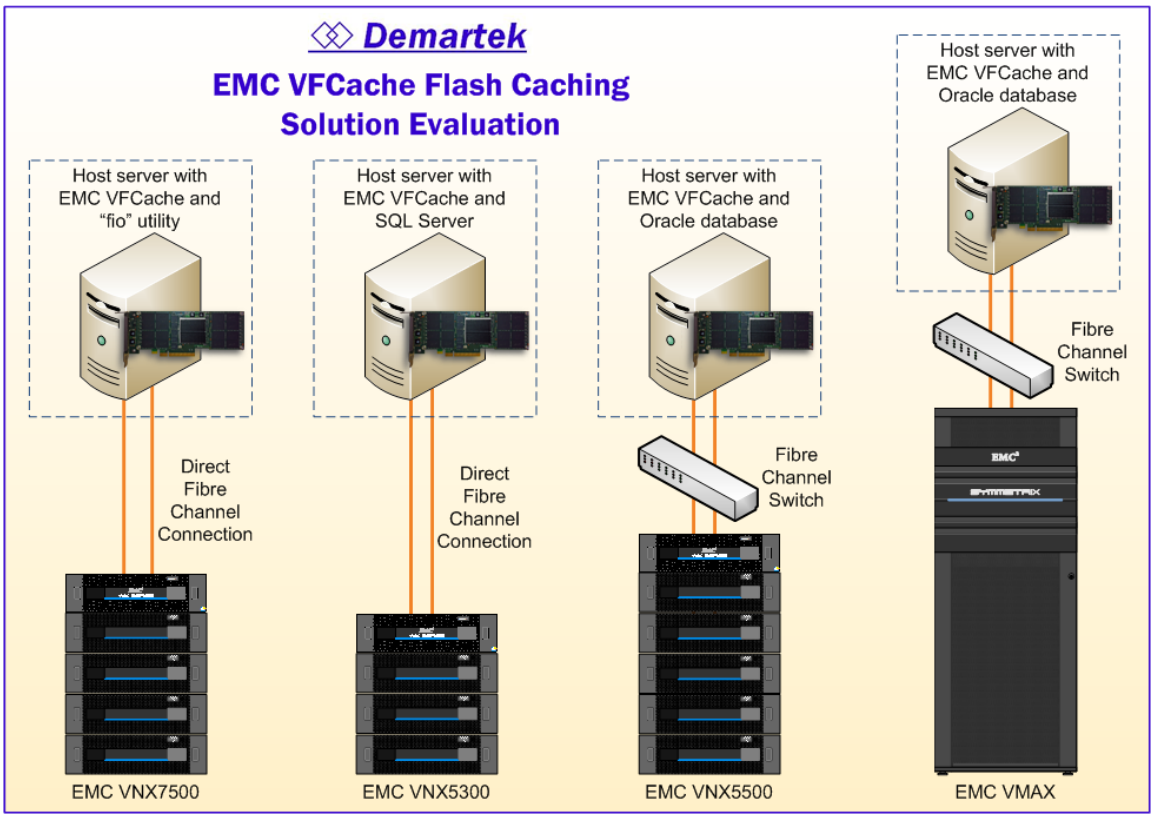
VFCache has the capability to split its SSD media into a cache section and a primary storage section. This allows administrators to allocate some of the NAND Flash to caching functions as described previously and allows a portion of the NAND Flash capacity to be directly used by applications that might need dedicated SSD storage. This primary storage capacity can be a nice complement to the caching portion of the SSD. It should be noted that normal data protection procedures will need to be applied to any data that is stored in the primary storage capacity portion of the VFCache card.

Performance Test Results

A series of performance benchmarks were run on four server and storage combinations, with each server equipped with EMC VFCache. These tests included OLTP benchmarks that are similar to TPC-C and TPC-E workloads. They were run in various configurations, using Oracle and Microsoft SQL Server database environments. A separate test using "fio" on a Linux host was also run. The tool "fio" can be used for benchmark and I/O stress verification purposes.

Each of these configurations provides enough spindles within the storage systems to allow the servers to drive reasonable workloads through the storage systems and to show the effect of VFCache on these workloads. However, these storage configurations are somewhat constrained in order to show the effect of VFCache on single applications. The VMAX and VNX storage systems can scale to much larger configurations and higher performance than those used in these particular tests. For example, the VMAX 4 engine storage system can scale to more than 1000 drives.

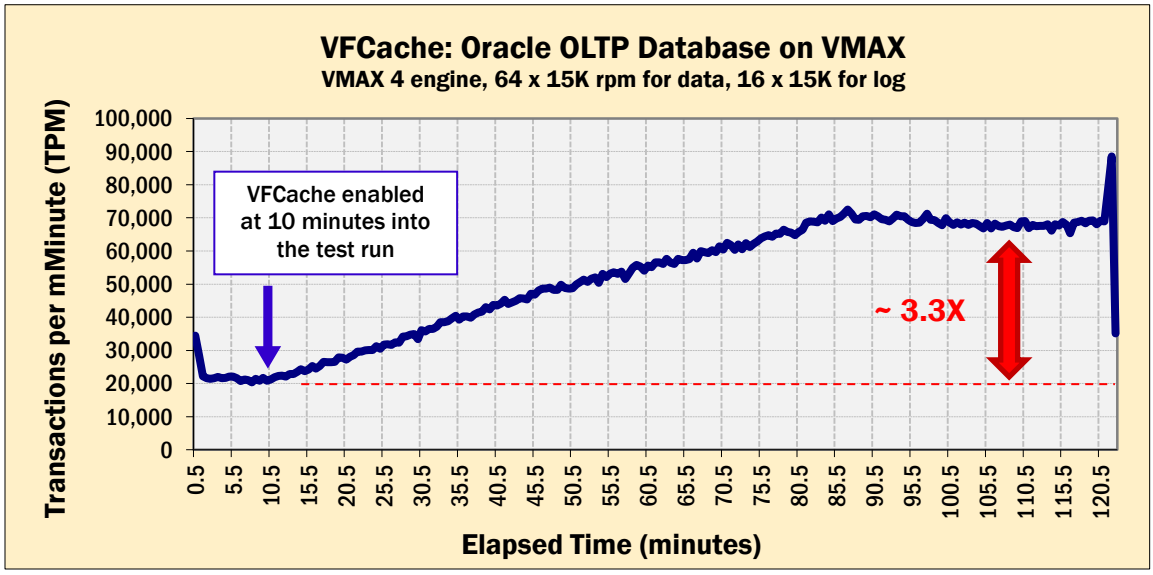
A diagram of the test configurations is shown below.



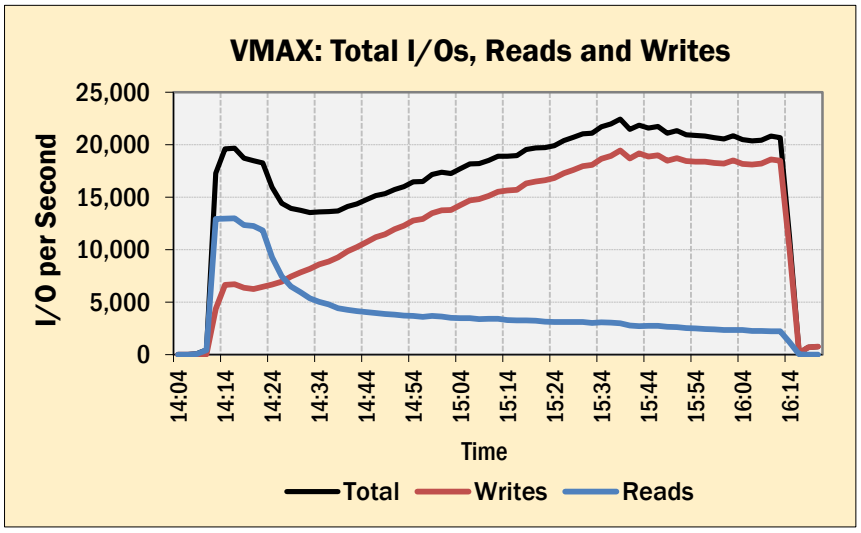
Oracle OLTP (TPC-C like) Workloads

An Oracle OLTP workload was run in two different configurations, one with a VMAX storage system and one with a VNX storage system. In both cases, a TPC-C like workload with approximately 70 percent reads and 30 percent writes was run on the host server for two hours, connected via Fibre Channel to the storage system. We first observed the tests running with VFCache disabled looking for stable performance. For these tests, the performance without VFCache stabilized relatively quickly, so we decided that after ten minutes, we would enable VFCache.

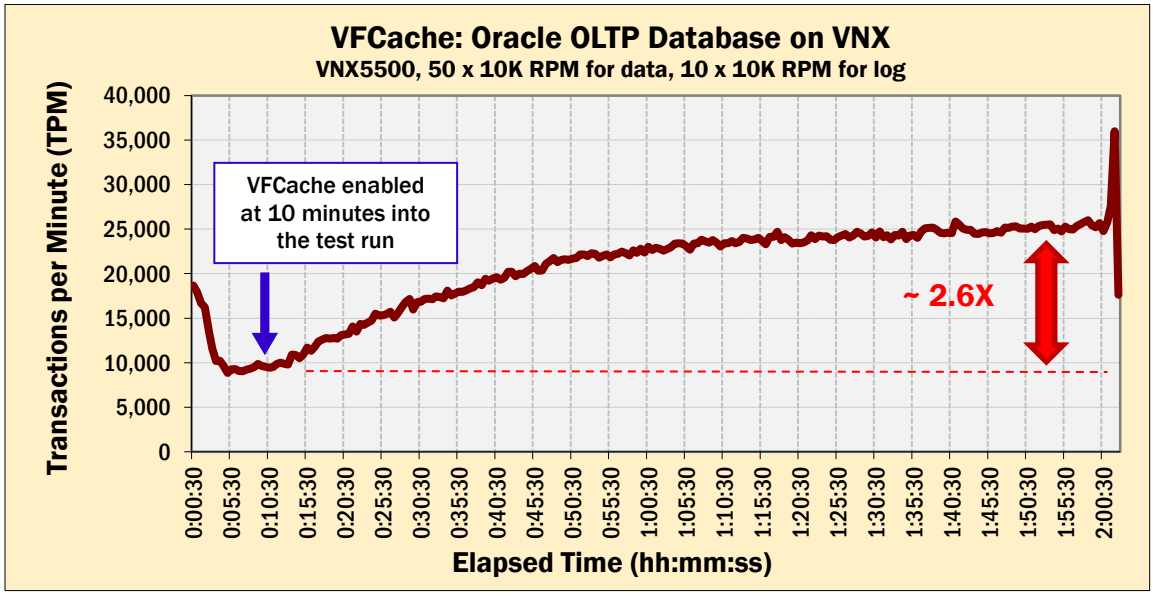
The graphs show the effect of the SSD caching, showing the normal cache "warm-up" as the cache began to fill and accelerate the performance of the application workload. For this test, the full performance of more than 3 times performance was achieved after VFCache had been enabled for approximately 75 minutes. The system performance doubled in approximately 25 minutes.



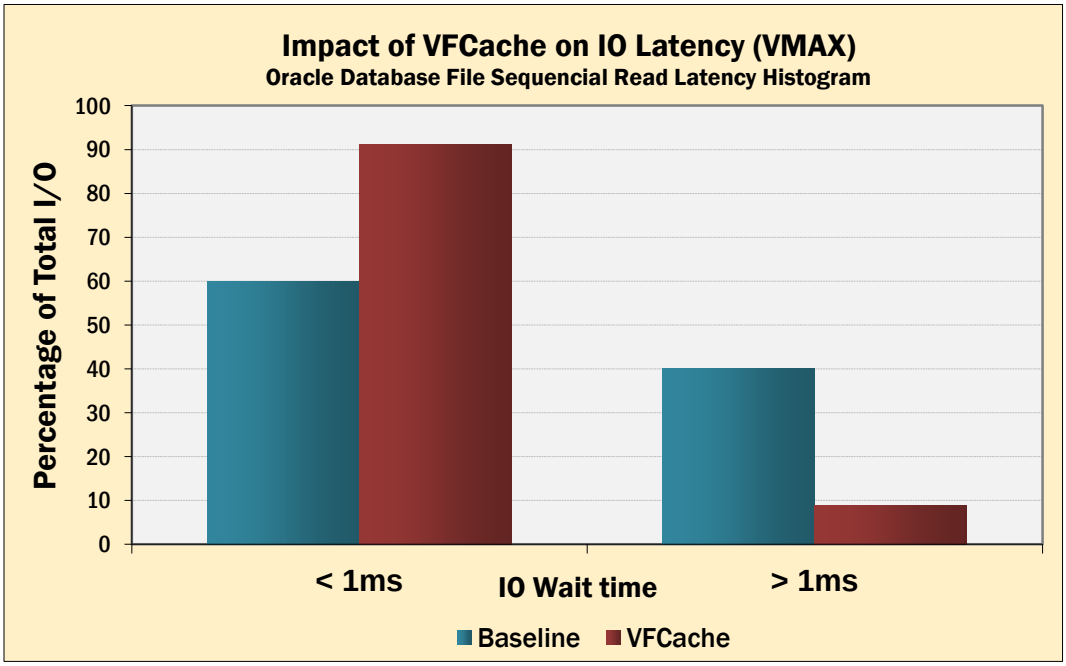
The effect of VFCache on the VMAX can be observed by viewing the I/O statistics from the VMAX during this test. As the VFCache began to fill with read data, fewer reads were sent to the VMAX, allowing the VMAX to complete writes faster, improving overall performance.



A similar performance graph for the Oracle OLTP workload on the VNX was observed, showing significant performance improvement for the application workload.



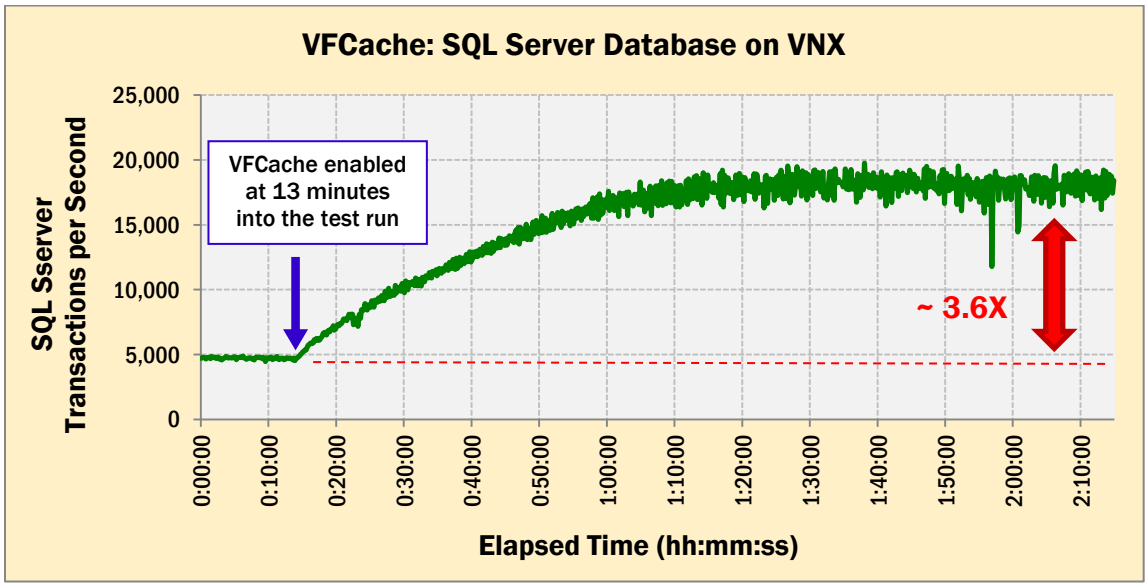
VFCache had a positive effect on Oracle Wait Events by decreasing the overall latency and causing more than 90% of the wait events to be satisfied in less than one millisecond (< 1 ms) as compared to approximately 60% without VFCache.



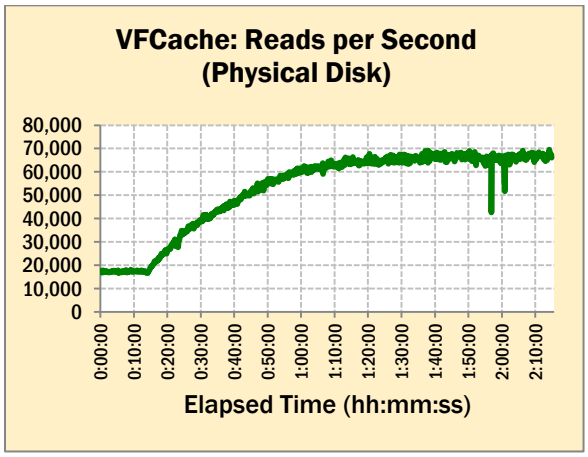
Microsoft SQL Server OLTP (TPC-E like) Workload

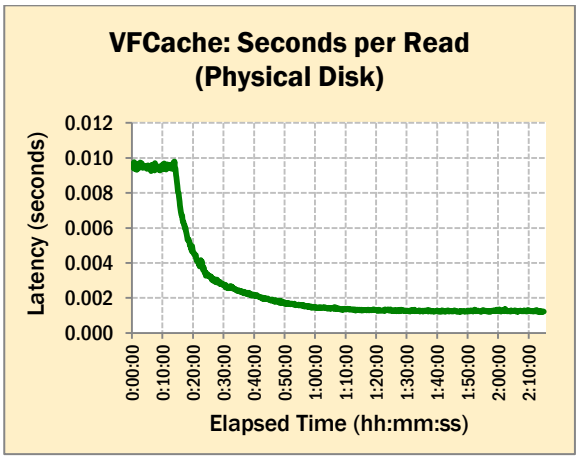
A Microsoft SQL Server OLTP workload was run on a server connected to an EMC VNX storage system. This TPC-E like workload with a high read mix was run on the host server for slightly more than two hours, connected via Fibre Channel to the EMC storage system. We first observed the tests running with VFCache disabled looking for stable performance. For these tests, the performance without VFCache stabilized relatively quickly, so we decided that after thirteen minutes, we would enable VFCache.

We captured data from the native Windows Performance Monitor (Perfmon) including SQL Server database transactions per second and various physical disk statistics. A similar performance improvement and cache warm-up effect was observed as with the other tests. Cache warm-up times will vary depending on workloads.



The two graphs below show the activity on the physical database disk as observed by the operating system. Note the significant read latency reduction when the cache was enabled.





Open Source "fio" Utility

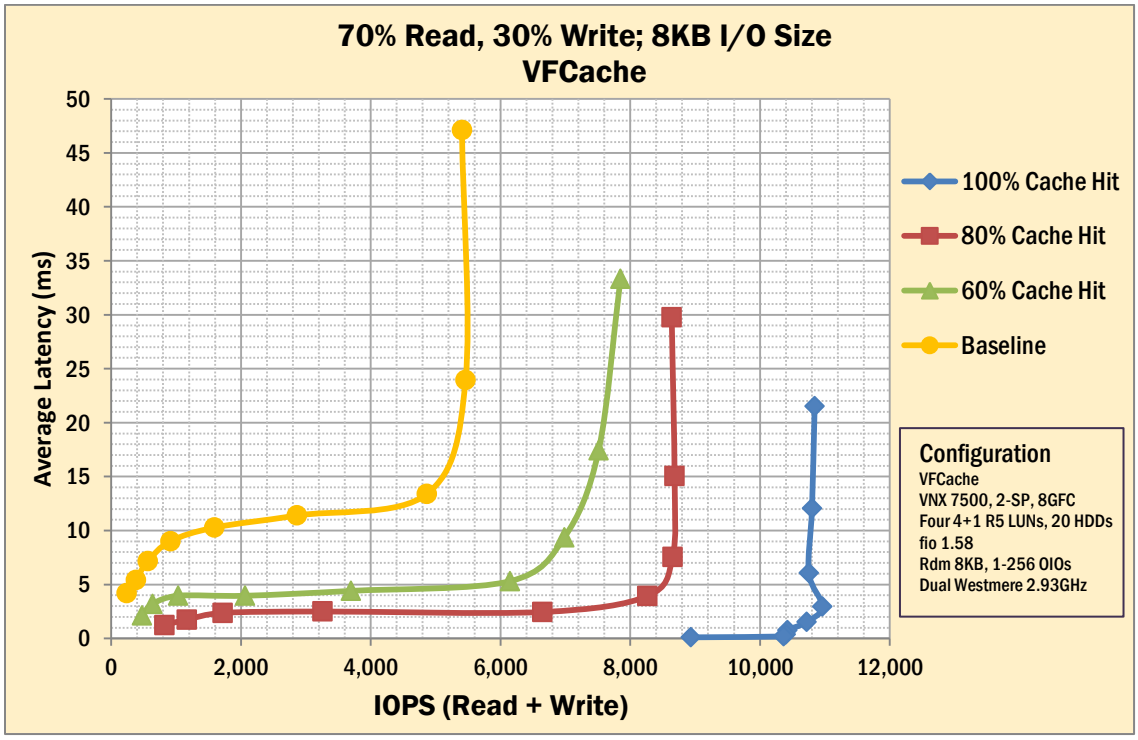
A fourth server and storage combination was used to show the response time (latency) at various cache hit levels for VFCache. For simplicity, the server used in this set of tests was directly connected via its Fibre Channel host interface without a Fibre Channel switch. For this test, the Linux I/O benchmark tool known as "fio" was used. This "fio" tool is an open-source utility that can be used for benchmark and I/O stress verification purposes. It was configured to run a read/write mix of 70 percent read and 30 percent write at 8KB block sizes and in such a way as to achieve a 60 percent cache hit rate, 80 percent cache hit rate and 100 percent cache hit rate. A baseline test was also run with the SSD caching disabled.

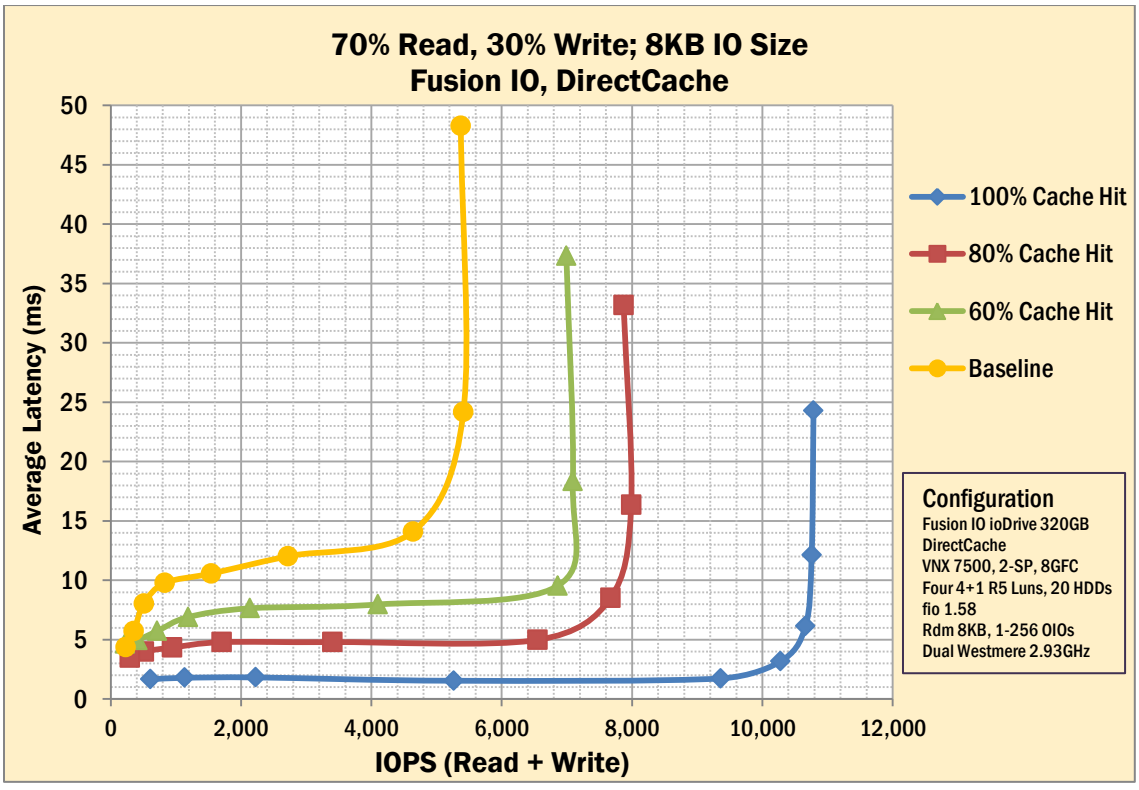
The test was configured to increase the outstanding I/Os, also known as queue depth, from one to 256 simultaneous I/O requests, and the results were plotted. As the cache hit rate is increased, the number of I/O operations per second (IOPS) increases and the latency decreases.

As the number of outstanding I/Os increases with each cache hit rate factor, eventually the caching solution will achieve its maximum I/O rate with only the latencies increasing.

To provide a point of comparison, VFCache was tested and then a competing solution from Fusion-io with its DirectCache software was installed and the test repeated. The SSD caches were approximately the same size of 320GB raw capacity.

In these tests, VFCache showed better performance and lower latency than the Fusion-io solution as the cache hit rate increased. The VFCache solution was able to achieve higher IOPS while at the same time providing better (lower) response time (latency) at each number of outstanding I/Os than the Fusion-io solution. The VFCache solution performed better than the Fusion-io solution because the combination of the VFCache PCIe card and its software operate faster than the Fusion-io PCIe card with its DirectCache software.


70% Read, 30% Write; 8KB I/O Size
VFCache
Average Latency (ms)
IOPS (Read + Write)
0
5
10
15
20
25
30
35
40
45
50
0
2,000
4,000
6,000
8,000
10,000
12,000
100% Cache Hit
80% Cache Hit
60% Cache Hit
Baseline
Configuration
VFCache
VNX 7500, 2-SP, 8GFC
Four 4+1 R5 LUNs, 20 HDDs
fio 1.58
Rdm 8KB, 1-256 OIOs
Dual Westmere 2.93GHz


70% Read, 30% Write; 8KB IO Size
Fusion IO, DirectCache
Average Latency (ms)
IOPS (Read + Write)
0
5
10
15
20
25
30
35
40
45
50
0
2,000
4,000
6,000
8,000
10,000
12,000
100% Cache Hit
80% Cache Hit
60% Cache Hit
Baseline
Configuration
Fusion IO ioDrive 320GB
DirectCache
VNX 7500, 2-SP, 8GFC
Four 4+1 R5 Luns, 20 HDDs
fio 1.58
Rdm 8KB, 1-256 OIOs
Dual Westmere 2.93GHz

Conclusion

As we have shown with these tests, the VFCache Flash caching solution provided from 2.5 times to 3.5 times performance gain with the OLTP database workloads, while reducing the latency for these transactions. While not every workload will achieve these exact results, this shows the type of strong performance gains possible with VFCache.

By using VFCache, we have also demonstrated that as the cache warms-up, the reduced read load on the back-end storage system allows that storage system to increase its write throughput, increasing overall performance. We also observed reduced latencies in the form of very fast wait event times for the Oracle database and significantly reduced read latency for Microsoft SQL Server.

We also were able to show the VFCache Flash caching solution outperforming a competitive solution.

When VFCache is deployed from a single server, the backend storage connected to that server will experience reduced utilization because the server cache is handling a large number of reads. This also benefits other servers that may be connected to that storage system, because there will be more I/O resource available to the other servers. If VFCache is deployed in multiple servers, then the cache benefit is multiplied across those servers further reducing the load on the backend storage systems.

Even if a server with VFCache does not require increased I/O workload for its backend storage, the server and the applications running in it will benefit by having significantly reduced latency for the existing reads, improving application performance.

VFCache is a very cost-effective way to improve the performance of heavy-duty read-intensive I/O workload applications and reduce the overall load on backend storage systems.

Appendix – Evaluation Environment

Four separate sets of servers and storage were used for these tests. For each test, one server was equipped with VFCache and connected to one EMC storage system – via Fibre Channel. Three of these servers were connected to various models of EMC VNX storage systems. One server was connected to an EMC VMAX storage system.

Oracle workload server #1

Server: Cisco C460 M1
Operating System: RedHat Enterprise Linux 5 U7
Switches: Brocade DS-5300B (two, in redundant configuration)
Storage: EMC VMAX, FAST VP disabled

Oracle Workload Profile
Oracle 11gR2 (11.2.0.1) with ASM for volume management
Database size: 1.2 TB
OLTP benchmark: TPC-C like workload with 5000 warehouses and 100 concurrent users
Read / Write ratio: 70 / 30
ASM DATA diskgroup: 64x 450GB 15K RPM FC disk drives defined as 16x 200GB LUNs
ASM REDO diskgroup: 16x 450GB 15K RPM FC disk drives defined as 4x 250GB LUNs

Oracle workload server #2

Server: Cisco C250 M1
Operating System: RedHat Enterprise Linux 5 U7
Switches: Brocade DS-5300B (two, in redundant configuration)
Storage: EMC VNX5500, FAST Suite disabled

Oracle Workload Profile
Oracle 11gR2 (11.2.0.3) with ASM for volume management
Database size: 1.2 TB
OLTP benchmark: TPC-C like workload with 5000 warehouses and 100 concurrent users
Read / Write ratio: 70 / 30
ASM DATA diskgroup: 50x 600GB 15K RPM SAS disk drives defined as 10x 250GB LUNs
ASM REDO diskgroup: 10x 600GB 15K RPM SAS disk drives defined as 4x 200GB LUNs

SQL Server workload server

Server: Cisco UCS C460 M1
Operating System: Windows Server 2008 R2 SP1
Switches: none, this server was directly connected to the storage
Storage: EMC VNX5300, FAST Suite disabled

SQL Server Workload Profile
SQL Server 2008 R2
Database size: 1.3TB
OLTP benchmark: TPC-E like workload with 90,000 warehouses and 120 concurrent users

Open Source "fio" Utility workload server

Server: 2x Intel Xeon X5570, 2.93GHz, 12 total cores
Operating System: CentOS 5.6
Switches: none, this server was directly connected to the storage
Storage: EMC VNX7500, FAST Suite disabled

This report is available at http://www.demartek.com/VFCache on the Demartek web site.