

EMC VNX2 Multicore FAST Cache

VNX5200, VNX5400, VNX5600, VNX5800, VNX7600, & VNX8000
A Detailed Review

Abstract

This white paper is an introduction to the EMC[®] Multicore FAST[™] Cache technology in the VNX[®]2 storage systems. It describes implementation of the Multicore FAST[™] Cache feature and provides details of using it with Unisphere[®] and NaviSecCLI. Usage guidelines and major customer benefits are also included.

March 2016

Copyright © 2016 EMC Corporation. All Rights Reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided “as is.” EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

VMware and VMware View are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. All other trademarks used herein are the property of their respective owners.

Part Number H12208.4

Table of Contents

Executive Summary	4
Introduction	4
Audience	5
Terminology	5
Global Multicore FAST Cache and TCO	6
Multicore FAST Cache Components	7
Theory of Operation	7
Multicore FAST Cache Promotions	7
Host Read Operations	8
Host Write Operations	9
Multicore Cache Copies to Multicore FAST Cache	10
Multicore FAST Cache Cleaning	11
Multicore FAST Cache Flush	11
Management	11
Unisphere	12
NaviSecCLI	16
Unisphere Analyzer	16
Best Practices	17
Interoperability Considerations	17
Failure Handling	18
Conclusion	19
References	19
Appendix A Multicore FAST Cache Configuration Options	20
Appendix B FAST VP and Multicore FAST Cache	21
Appendix C Multicore FAST Cache Comparison with Storage System Cache	23

Executive Summary

Since the original deployment of Flash technology in disk modules (commonly known as SSDs) in enterprise arrays, EMC® has expanded the use of this technology throughout the EMC® VNX® series. The combination of high performance and the rapidly falling cost-per-gigabyte of Flash technology led to the concept of a caching tier. A caching tier is a large-capacity secondary cache that uses FAST Cache Optimized Flash drives positioned between the storage processor's DRAM-based primary cache and hard-disk drives (HDD). On EMC VNX2 storage systems, this feature is called EMC Multicore FAST™ Cache.

Multicore FAST Cache extends the storage system's existing caching capacity for better system-wide performance. It achieves this by extending the functionality of the DRAM cache by copying frequently accessed data to FAST Cache Optimized Flash drives, which are faster than HDDs, therefore boosting system performance. Multicore FAST Cache also provides a much larger, scalable cache than the DRAM cache. The capacities range from 100 GB to 4.8 TB, which is considerably larger than the available DRAM cache of existing storage systems (see [Appendix A Multicore FAST Cache Configuration Options](#)).

At a system level, Multicore FAST Cache makes the most efficient use of Flash drive capacity. It achieves this by using Flash drives for the most frequently accessed data in the storage system instead of dedicating the drives to a particular application. Configuring Multicore FAST Cache is a non-disruptive online process that uses the existing memory-allocation interface but does not use host (server) cycles. It is created in RAID-protected mirrored pairs, and the capacity options depend on the storage-system model and the number and type of installed Flash drives. You can create Multicore FAST Cache, enable it on storage volumes, and manage it through Unisphere®. No user intervention is required to enable applications to see the performance benefits of Multicore FAST Cache. Multicore FAST Cache can be used for Classic LUNs and Pool LUNs.

Introduction

This white paper provides an introduction to the Multicore FAST Cache feature. The purpose of Multicore FAST Cache is to leverage the performance of Flash drives and efficiently use them to store the most highly accessed data within the system. If a particular chunk of data is accessed frequently, the VNX2 storage system automatically copies that chunk into Multicore FAST Cache from the hard disk drives into Flash drives. Subsequent I/O access to the same chunk is serviced at Flash-drive response times, thus boosting storage-system performance. If the access frequency of this chunk of data decreases and other chunks need to be copied to Multicore FAST Cache, the least recently used data is replaced first. Details about the Multicore FAST Cache algorithm and basic operations are discussed in the [Theory of Operation](#) section of this white paper.

Audience

This white paper is intended for EMC customers, partners, and employees who are considering the use of the Multicore FAST Cache feature in VNX2 storage systems. It assumes familiarity with VNX2 storage systems and EMC's management software.

Terminology

- **Cache page**—The smallest unit of allocation inside the Multicore FAST Cache, 64 KB in size.
- **Cache clean page**—A page of Multicore FAST Cache that is valid and contains a copy of data that has been synchronized with the user LUN.
- **Cache dirty page**—A page of Multicore FAST Cache that is valid and contains the most recent copy of data, but has not yet been synchronized with the user LUN.
- **Cache warming**—The process of copying new pages into FAST Cache after it has been created, or a change in the application access profile that begins to reference an entirely new set of data.
- **Chunk**—A portion of data in a particular address range (64 KB).
- **DRAM memory**—A storage-system component used by Multicore Cache to store data in very fast storage media (DRAM), to serve requests for that data faster.
- **Extent**—A set of adjacent physical blocks.
- **Multicore FAST Cache copy**—The process of copying data from a Multicore FAST Cache page to a back-end hard-disk-based LUN.
- **Multicore FAST Cache promotion**—The process of copying data from an HDD to a Multicore FAST Cache page.
- **Multicore FAST Cache flush**—The process of copying data from a Multicore FAST Cache page to a back-end hard-disk-based LUN freeing the page for use.
- **Multicore FAST Cache Hit**—The instance when an I/O can be completed using the contents of Multicore FAST Cache.
- **Multicore FAST Cache Miss**—The instance when Multicore FAST Cache does not contain the data needed to complete the I/O and the HDD must be used.
- **Flash drive**—A data storage device that uses solid-state media to store data. Because it does not have moving parts, a Flash drive provides extremely low response times and high IOPS compared to rotating hard-disk drives (HDDs).
- **Hard disk drive (HDD)**—A data storage device that stores data on magnetic surfaces and rotates at various speeds.
- **Hot spot**—A busy area on a LUN.
- **Locality of reference**—The concept that logical blocks located close to each other are accessed at approximately the same time and repeatedly.
- **Logical block address**—An addressing scheme that specifies the location of blocks of data on storage devices.
- **Memory map**—An array of addresses in which each bit represents a Multicore FAST Cache page. This map shows which pages are in Multicore FAST Cache and where they are located. A copy of the memory map resides in DRAM cache. This ensures that the pages are accessed at memory speeds.

- **Multicore Cache**—An MCx software component that increases host write and read performance by using the VNX2 storage processor's DRAM.
- **Multicore Cache Hit**—The instance when an I/O can be completed using the contents of Multicore Cache.
- **Multicore Cache Miss**—The instance when Multicore FAST Cache does not contain the data needed to complete the I/O.
- **Pool**—A group of disk drives used by pool LUNs. There may be zero or more pools on a system. Disks may only be a member of one pool. Pool disks cannot be used in RAID groups.
- **Thin LUN**—A logical unit of storage created on a pool where physical space consumed by the storage system may be less than the user capacity seen by the host server.
- **Thick LUN**—A logical unit of storage created on a pool where physical space consumed on the storage system is equal to the user capacity seen by the host server.

Global Multicore FAST Cache and TCO

Multicore FAST Cache allows you to leverage the lower response time and better IOPS of Flash drives without dedicating Flash drives to specific applications. This technology supplements the available storage-system cache (adding up to 4.8 TB read/write Multicore FAST Cache in the VNX7600 and VNX8000 storage systems; see [Appendix A Multicore FAST Cache Configuration Options](#)). Multicore FAST Cache addresses a hot spot anywhere in the array, either on Classic LUNs or storage pool LUNs.

One of the major benefits of using Multicore FAST Cache is the improved application performance, especially for workloads with frequent and unpredictable large increases in I/O activity. The part of an application's working dataset that is frequently accessed is copied to the Multicore FAST Cache, so the application receives an immediate performance boost. Multicore FAST Cache enables applications to deliver consistent performance by absorbing bursts of read/write loads at Flash drive speeds.

Another important benefit is improved total cost of ownership (TCO) of the system. Multicore FAST Cache copies the hot or active subsets of data to Flash drives in chunks. By offloading many if not most of the remaining IOPS after Multicore Cache, the customer can fill the remainder of their storage needs with low cost, high capacity disk drives. This ratio of a small amount of Flash paired with a lot of disk offers the best performance (\$/IOPS) at the lowest cost (\$/GB) with optimal power efficiency (IOPS/KWH).

VNX OE for Block version 05.33.009.5.155 introduces the ability to utilize SAS Flash 2 drives in Multicore FAST Cache. In previous VNX OE for Block versions, only SAS Flash drives are supported. By utilizing SAS Flash 2 drives in Multicore FAST Cache, the TCO of the system is further reduced. Multicore FAST Cache can only be created with SAS

Flash drives or SAS Flash 2 drives. To change from one drive technology to another or to expand Multicore FAST Cache, Multicore FAST Cache must be disabled first.

Multicore FAST Cache is enabled by default on all Classic LUNs and Storage Pools once the FAST Cache enabler is installed. Classic LUNs and storage pools created before the enabler is installed will have Multicore FAST Cache disabled. To use Multicore FAST Cache for these items, you must manually enable Multicore FAST Cache by using either Unisphere or NaviSecCLI.

Multicore FAST Cache Components

Multicore FAST Cache requires the FAST Cache enabler to take advantage of the feature. To create Multicore FAST Cache, you need at least 2 FAST Cache Optimized drives in the system, which will be configured in RAID 1 mirrored pairs. Once the enabler is installed, the system uses the following main components to process and execute Multicore FAST Cache:

Policy Engine—Manages the flow of I/O through Multicore FAST Cache. When a chunk of data on a LUN is accessed frequently, it is copied temporarily to Multicore FAST Cache (FAST Cache Optimized drives). The Policy Engine also maintains statistical information about the data access patterns. The policies defined by the Policy Engine are system-defined and cannot be changed by the user.

Memory Map—Tracks extent usage and ownership in 64 KB chunks of granularity. The Memory Map maintains information on the state of 64 KB chunks of storage and the contents in Multicore FAST Cache. A copy of the Memory Map is stored in DRAM memory, so when the FAST Cache enabler is installed, SP memory is dynamically allocated to the Multicore FAST Cache Memory Map. The size of the Memory Map increases linearly with the size of Multicore FAST Cache being created. A copy of the Memory Map is also mirrored to the Flash disks to maintain data integrity and high availability of data.

Theory of Operation

Multicore FAST Cache Promotions

A Multicore FAST Cache promotion is the process in which data is copied from spinning media HDDs and placed into Multicore FAST Cache. The process is defined as a promotion due to the upgrade in performance the data receives from being copied from spinning media to Flash technology.

Causes of a Multicore FAST Cache Promotion

During normal operation, a promotion to Multicore FAST Cache is initiated after the Policy Engine determines that 64 KB block of data is being accessed frequently. To be considered, the 64 KB block of data must be accessed by reads and/or writes multiple times within a short period of time. The data also cannot reside on Flash based storage already, which means data currently residing on an Extreme Performance tier of a pool or on a Flash based Classic RAID group are not eligible to be copied into Multicore FAST Cache. This restriction exists because the data would not benefit moving from Flash drives to other Flash drives. For information on how

FAST™ VP and Multicore FAST Cache work together, see [Appendix B FAST VP and Multicore FAST Cache](#).

Promotion Operation

After the Policy Engine determines the 64 KB block of data should be promoted to Multicore FAST Cache due to one of the above scenarios, the 64 KB region is copied from the HDD to Multicore FAST Cache. The Memory Map is then updated to indicate that the data now resides in Multicore FAST Cache. When the application accesses this data again and a Multicore Cache Miss is encountered, the I/O is directed to Multicore FAST Cache. This is called a Multicore FAST Cache hit. Because the data is now accessed from the Flash drives, the application gets very low response times and high IOPS. If a substantial part of the working set is copied to Multicore FAST Cache over time, applications can see higher average performance, even with lower performing HDDs in the back end.

Access Patterns Not Considered For Promotion

Multicore Cache complements Multicore FAST Cache by handling certain I/O patterns that may not benefit from the use of Multicore FAST Cache. For example, small-block sequential I/O is handled at the Multicore Cache level, which means cycles are not spent copying data into Multicore FAST Cache that may not be used again. For example, small-block sequential writes will coalesce into larger I/Os to the HDDs, and small-block sequential reads will cause prefetching to occur. In both cases, larger back-end I/Os are created which may not cause the data being accessed to be promoted into Multicore FAST Cache. Multicore Cache also handles high-frequency access patterns, zero fill requests, and I/Os that are larger than 128 KB in size. Multicore Cache acts as a filter for Multicore FAST Cache, which provides more cycles to copy data into Multicore FAST Cache that will benefit from the use of it.

Host Read Operations

Incoming read I/O from a host application is first checked against the contents of Multicore Cache. If the I/O can be serviced from Multicore Cache, the I/O is completed. The Multicore FAST Cache Memory Map is not accessed during this exchange.

If a Read miss is encountered in Multicore Cache the Multicore FAST Cache Memory Map is checked to determine whether the I/O is for a chunk that already exists in Multicore FAST Cache. If the data exists in Multicore FAST Cache, the Policy Engine redirects the I/O request to Multicore FAST Cache. The data is then copied from Multicore FAST Cache to Multicore Cache. Multicore Cache then satisfies the read.

If the data is not in Multicore FAST Cache, the I/O request follows the same path it would follow if the storage system does not have Multicore FAST Cache. The data is copied from the HDDs into Multicore Cache. Multicore Cache satisfies the read request at this time. For frequently used data, the Policy Engine promotes the data into Multicore FAST Cache. [Figure 1](#) shows the read operation.

1. Host I/O is received, Multicore Cache checks if the I/O can be serviced from Multicore Cache
2. I/O request satisfied from Multicore Cache if data requested is in Multicore Cache
3. If a Read Miss is seen, Memory Map is checked to see if I/O is in Multicore FAST Cache
4. If I/O in Multicore FAST Cache, data is read from Multicore FAST Cache and placed in Multicore Cache, Multicore Cache completes the I/O
5. If Multicore FAST Cache miss is seen, data is read from HDD and placed in Multicore Cache, Multicore Cache completes the I/O
6. Multicore FAST Cache Policy Engine promotes the page to Multicore FAST Cache and updates the Memory Map if needed

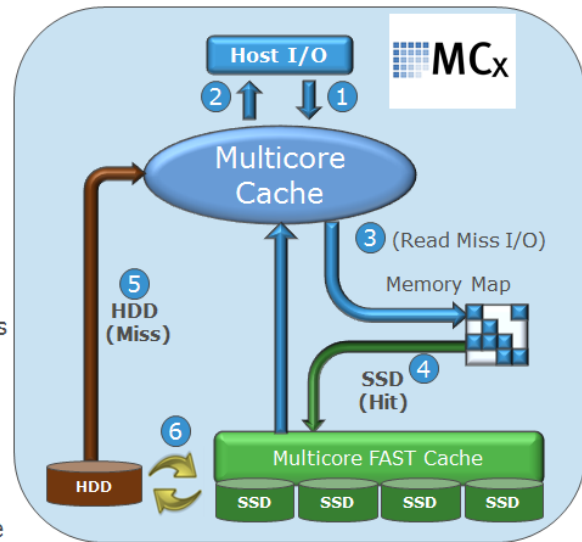


Figure 1. Multicore FAST Cache Read operation

Host Write Operations

If the host I/O request is a write operation, and Write Cache is enabled on the system and LUN, Multicore Cache services the I/O and sends an acknowledgment to the host. This exchange happens whether or not the data being updated is located in Multicore FAST Cache. The Memory Map is not accessed during this exchange.

In the instance where Write Cache is disabled on the system or LUN, and Multicore FAST Cache is enabled, Multicore Cache temporarily holds the I/O and checks the Memory Map to see if the data exists in Multicore FAST Cache. This is a Multicore Cache write-through operation. If the data being updated is in Multicore FAST Cache, Multicore Cache updates the data in Multicore FAST Cache and the Memory Map, and then completes the I/O by sending an acknowledgment to the host. If the data is not contained within Multicore FAST Cache, Multicore Cache updates the data on the LUN's underlying storage. For frequently used data, the Policy Engine copies the data into Multicore FAST Cache as a clean cache page. [Figure 2](#) further depicts this scenario.

1. Multicore Cache receives the write operation, the data must be saved to the storage before a write acknowledgement is returned
2. Memory Map is checked to see if the page is currently located in Multicore FAST Cache
3. If the page is located in Multicore FAST Cache, then the page is updated in Multicore FAST Cache
4. If the page is not located in Multicore FAST Cache, the data is saved to HDD
5. Multicore Cache acknowledges the write after the data is saved either to Multicore FAST Cache or the HDD
6. Multicore FAST Cache Policy Engine promotes the page to Multicore FAST Cache and updates the Memory Map if the data is being used frequently

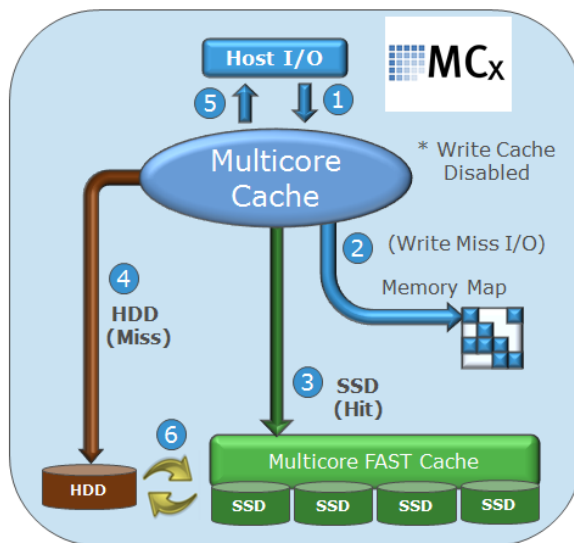


Figure 2. Multicore Cache write-through operation (Write Cache Disabled)

Multicore Cache Copies to Multicore FAST Cache

Data being copied into Multicore FAST Cache can also come directly from Multicore Cache (Figure 3). When Multicore Cache is cleaning dirty cache pages in DRAM, the Policy Engine is checked to see if the block being cleaned resides in Multicore FAST Cache. If it does, Multicore Cache copies the data from DRAM into Multicore FAST Cache directly and marks the page/s in DRAM as clean. The Multicore FAST Cache page is then marked as dirty. If the data does not reside in Multicore FAST Cache, the appropriate HDD is updated with the data being cleaned. At this point, the Policy Engine determines if the block being updated is being accessed frequently and is eligible for promotion. If it is, the data is also copied from DRAM into Multicore FAST Cache. Due to having exact copies of the data in Multicore FAST Cache and on the HDD, the Multicore FAST Cache page is considered clean. This means this type of copy to Multicore FAST Cache does not count against the Dirty Pages count for Multicore FAST Cache.

Note: The FAST Cache dirty pages (%) statistic only represents the percentage of data in Multicore FAST Cache which has not yet been copied back to the HDDs. This statistic does not represent the amount of data currently contained within Multicore FAST Cache.

1. Memory Map is checked to see if the page is currently located in Multicore FAST Cache
2. If the page is located in Multicore FAST Cache, then the Multicore Cache page is copied to Multicore FAST Cache
3. If the page is not located in Multicore FAST Cache, the data is copied to HDD
4. Multicore FAST Cache Policy Engine copies the page to Multicore FAST Cache as a clean page and updates the Memory Map if the data is being used frequently

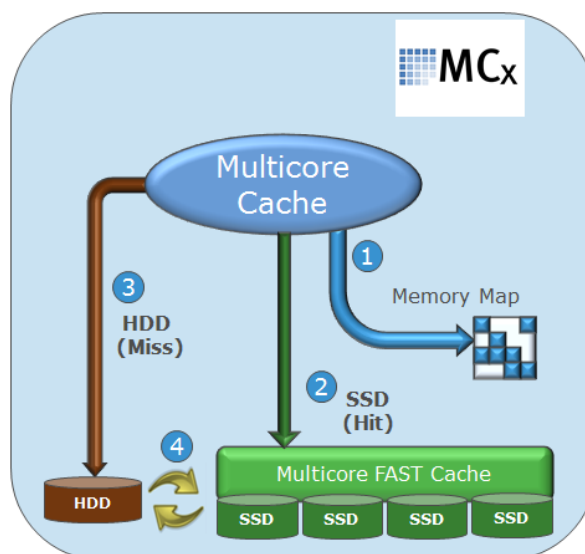


Figure 3. Multicore Cache copy operation

Multicore FAST Cache Cleaning

Multicore FAST Cache contains a cleaning process which proactively copies dirty pages to the underlying physical devices during times of minimal backend activity. During this cleaning process, the data is retained in Multicore FAST Cache so that subsequent host requests are still served by Multicore FAST Cache if the page(s) are accessed again. After a dirty page has been synchronized with the drives, the page becomes a clean cache page. Subsequent Multicore FAST Cache promotions use free pages first, then pages that have been cleaned on a least recently used basis before causing a Multicore FAST Cache dirty page to flush and be freed for a scheduled promotion. By having free and clean pages available, copies to Multicore FAST Cache are fast in comparison to flushing a dirty page for a scheduled promotion, which improves the performance of the system.

Multicore FAST Cache Flush

A Multicore FAST Cache Flush is the process in which a Multicore FAST Cache page is copied to the HDDs and the page is freed for use. In situations where no free or clean pages exist for a scheduled promotion, a Multicore FAST Cache flush occurs to accommodate this operation. The least recently used (LRU) algorithm determines which data blocks to flush to make room for the new promotions, and these pages are flushed from Multicore FAST Cache to the backend HDDs. The scheduled promotions are allowed to run as pages are freed.

Management

You can use Unisphere or NaviSecCLI to create, manage, and monitor Multicore FAST Cache. Unisphere details can be found in the *EMC Unisphere: Unified Storage Management Solution for the new VNX Series* white paper available on EMC Online Support. The following sections discuss the parts of Unisphere and NaviSecCLI that pertain to Multicore FAST Cache. For more information on Multicore FAST Cache configuration options, see [Appendix A Multicore FAST Cache Configuration Options](#).

Unisphere

The **System** tab in Unisphere has links on the right-hand task pane for System Properties and Manage Cache. Both these links open the System Properties window (Figure 4).

To enable Multicore FAST Cache, click the **FAST Cache** tab in the System Properties window to view Multicore FAST Cache information. If Multicore FAST Cache has not been created on the storage system, the **Create** button in the bottom of the dialog box is enabled. The **Destroy** button is enabled when Multicore FAST Cache has been created.

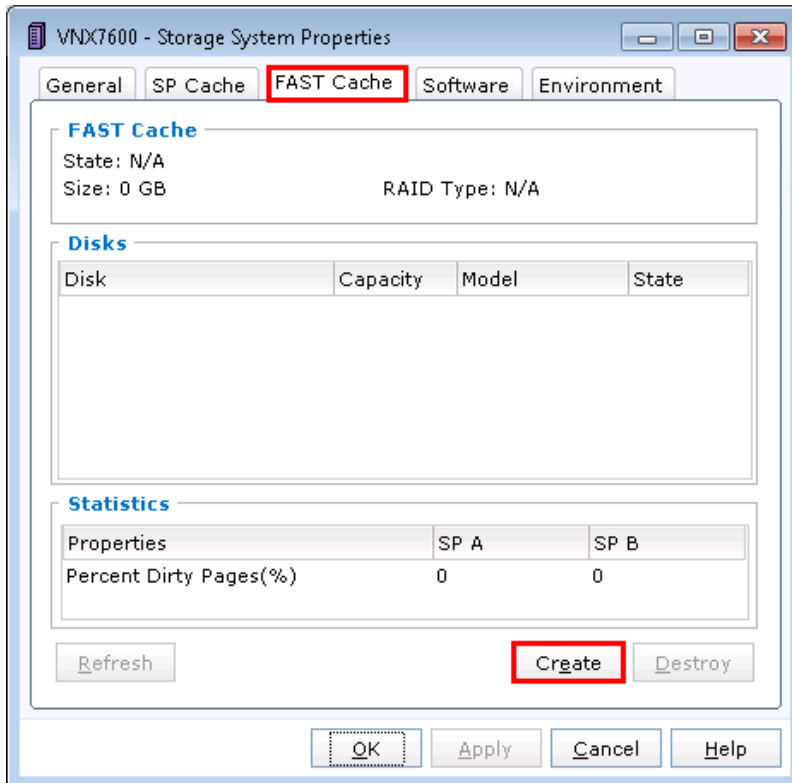


Figure 4. Storage System Properties dialog box

When Multicore FAST Cache has been created, the State, Size, and RAID Type fields are updated to reflect the configuration details. The RAID Type field displays **RAID 1** when Multicore FAST Cache has been created. Clicking **Create** opens the Create FAST Cache dialog box (Figure 5).

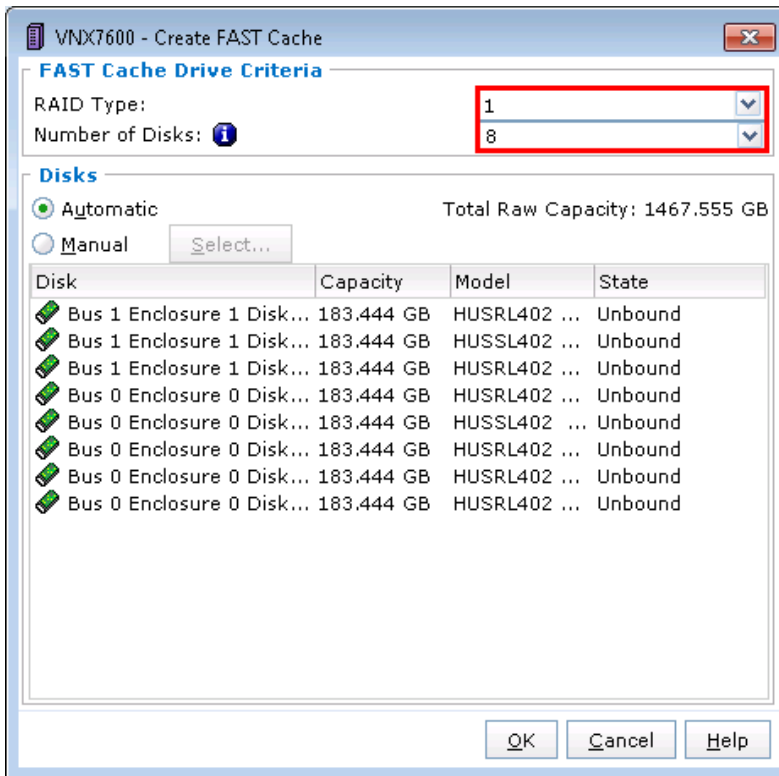


Figure 5. Create FAST Cache dialog box

If a sufficient number of Flash drives are not available to enable Multicore FAST Cache, Unisphere displays an error message, and Multicore FAST Cache cannot be created. The bottom portion of the screen shows the Flash drives that will be used for creating Multicore FAST Cache. You can choose the drives manually by selecting the **Manual** option. To change the size of Multicore FAST Cache after it is configured, you must destroy and recreate Multicore FAST Cache. This requires Multicore FAST Cache to flush all dirty pages currently contained in Multicore FAST Cache. When Multicore FAST Cache is created again, it must repopulate its data (warm-up period).

Figure 6 shows how you can enable Multicore FAST Cache for Classic LUNs under the **Advanced** tab in the Create LUN dialog box. If the Classic LUN has already been created, click the **Cache** tab in the **LUN Properties** dialog box to configure FAST Cache (Figure 7).

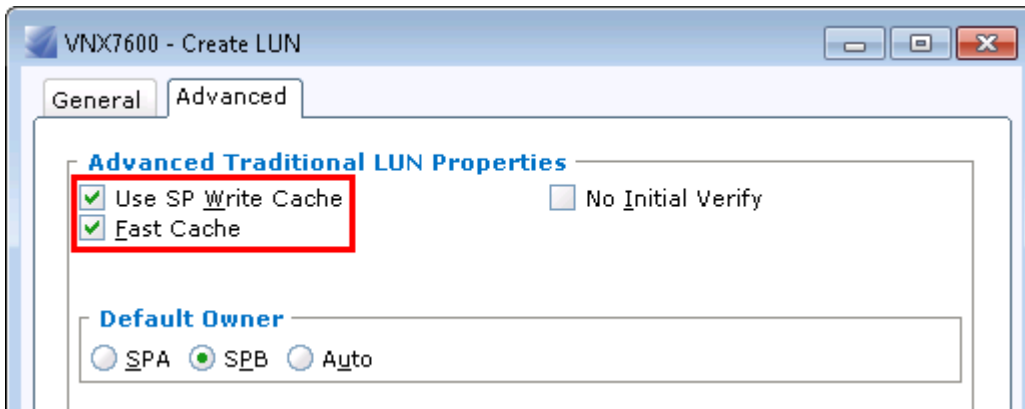


Figure 6. Advanced tab in the Create LUN dialog box (Classic LUNs)

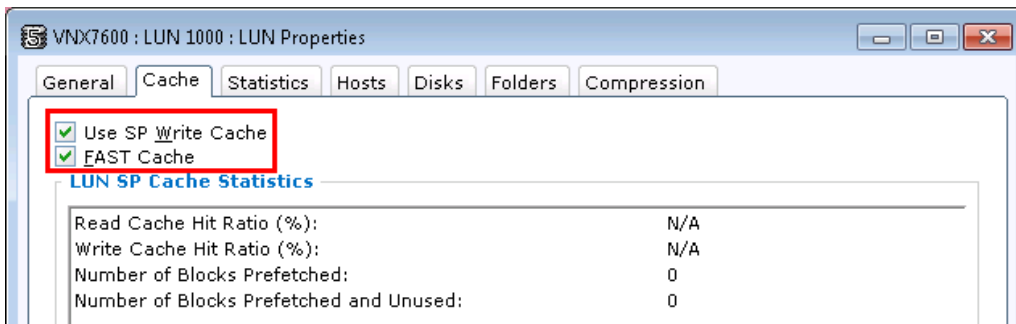


Figure 7. Cache tab in the LUN Properties dialog box

With storage pools, Multicore FAST Cache is enabled on a per pool basis. All the LUNs created in the storage pool will have Multicore FAST Cache enabled or disabled collectively. You can configure Multicore FAST Cache on Pools by using the **Advanced** tab in the Create Storage Pool dialog box shown in [Figure 8](#).

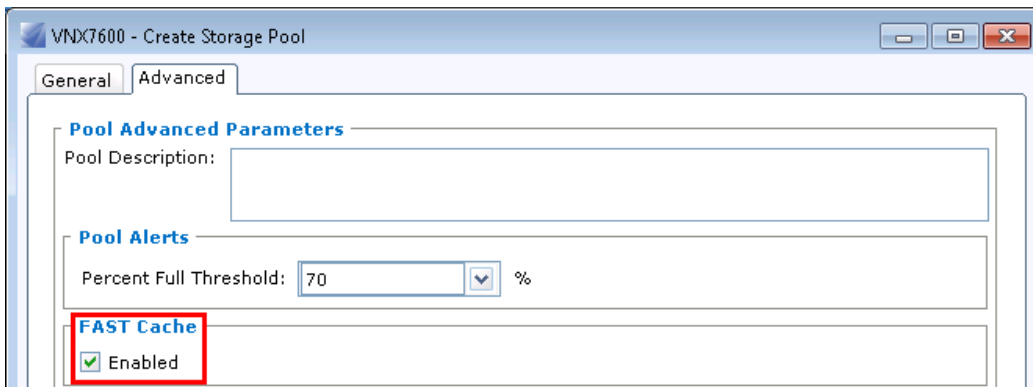


Figure 8. Advanced tab in the Create Storage Pool dialog box

If the storage pool has already been created, use the **Advanced** tab in the Storage Pool Properties dialog box to enable Multicore FAST Cache ([Figure 9](#)).

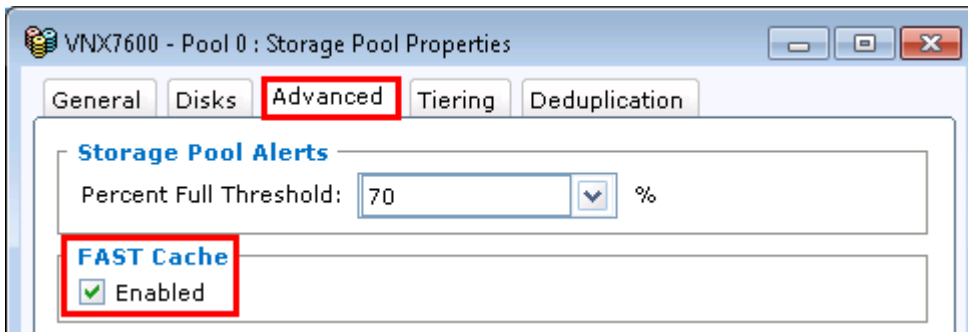


Figure 9. Advanced tab in the Storage Pool Properties dialog box

You can display Multicore FAST Cache properties in any Unisphere table (for example, the LUNs table) by right-clicking the table header and selecting **Choose Columns**. You can also click the Tools icon at the top-right corner of the table and select **Choose Columns**. This opens a dialog box, shown in [Figure 10](#), where you can select **FAST Cache**. The FAST Cache property is displayed for every entry in the table.

The screenshot shows the 'LUNs' table in Unisphere. The table has columns for Name, ID, RAID Type, FAST Cache, and User Capacity (GB). The 'FAST Cache' column is highlighted with a red box. The table contains four entries, all with 'On' status.

Name	ID	RAID Type	FAST Cache	User Capacity (GB)
FAST_Cache_1		0 RAID5	On	20.000
FAST_Cache_2		1 RAID5	On	20.000
FAST_Cache_3		2 RAID5	On	20.000
FAST_Cache_4		3 RAID5	On	20.000

Figure 10. Multicore FAST Cache column on the LUN screen in Unisphere

NaviSecCLI

The management functions described in the previous section are also available with NaviSecCLI. [Table 1](#) lists the CLI commands for Multicore FAST Cache:

Table 1. Multicore FAST Cache CLI commands

Task	NaviSecCli command
Create Multicore FAST Cache	cache –fast -create
Destroy Multicore FAST Cache	cache –fast -destroy
Get Multicore FAST Cache info	cache –fast -info
Configure Multicore FAST Cache when creating a Classic LUN	bind ... -fastcache 0 1
Enable of disable Multicore FAST Cache on a Classic LUN	chglun - <LUN#> -fastcache 0 1
Get Multicore FAST Cache config info on a Classic LUN	getlun <LUN#> -fastcache
Configure Multicore FAST Cache when creating a storage pool	storagepool –create ... -fastcache on off
Configure Multicore FAST Cache on existing storage pool	storagepool –modify –id <#> -fastcache on off
Get Multicore FAST Cache state on storage pool	storagepool –list –id <#> -fastcache

“...” is indicative that more cli options are required.

Unisphere Analyzer

Unisphere Analyzer gathers Multicore FAST Cache statistics to monitor performance. To view these statistics, enable Analyzer’s **Advanced** mode by using the following steps:

1. In Unisphere, click **System**.
2. Click **Monitoring and Alerts**.
3. Click **Statistics for Block**.
4. Click **Customize Charts**.
5. Click the **General** tab.
6. Select the **Advanced** checkbox.
7. Click **OK** to apply the settings.

The following Multicore FAST Cache statistics are available at the storage processor level:

- FAST Cache Dirty Pages (%)
- FAST Cache MBs Flushed (MB/s)

The following Multicore FAST Cache statistics are available for Classic LUNs and storage pools:

- FAST Cache Read Hits/s
- FAST Cache Read Misses/s
- FAST Cache Read Hit Ratio
- FAST Cache Write Hits/s

- FAST Cache Write Misses/s
- FAST Cache Write Hit Ratio

EMC Online Support contains videos that will help you view these statistics. Log into EMC Online Support and search for “Analyzer Series” for the Multicore FAST Cache video.

Best Practices

- Preferred application workloads for Multicore FAST Cache:
 - Small-block random I/O applications with high locality
 - Data re-hits—High frequency of access to the same data
 - Systems where current performance is limited by HDD capability, not SP capability
- If you have a limited number of Flash drives and an option to use them either for FAST VP or Multicore FAST Cache, EMC recommends that you use FAST Cache Optimized drives to create Multicore FAST Cache. Then use the remaining Flash drives in a FAST VP-enabled storage pool. Multicore FAST Cache is global in nature and benefits all the LUNs and pools in the storage system. FAST VP only benefits the storage pool where the Flash drives reside. For more information on Multicore FAST Cache and FAST VP, see [Appendix B FAST VP and Multicore FAST Cache](#).
- Unisphere allows you to choose FAST Cache optimized drives to use to create Multicore FAST Cache. You can also choose these drives manually to ensure that you distribute the Flash drives across back-end buses if desired.
- Multicore FAST Cache can improve overall system performance if the current bottleneck is drive-related, but boosting the IOPS will result in greater CPU utilization on the SPs. Systems should be sized so that the maximum sustained utilization is 70 percent. On an existing system, check the SP CPU utilization. If the utilization is over 80 percent, contact an EMC storage specialist to review the system health and determine next steps, before enabling Multicore FAST Cache.

Note: For storage pools, Multicore FAST Cache is a pool wide feature so you will enable/disable at the pool level (for all LUNs in the pool).

More detailed best practice guidelines can be found in the *VNX Unified Best Practices for Performance* white paper available on EMC Online Support. Refer to application-specific white papers for guidelines on using Multicore FAST Cache with those applications.

Interoperability Considerations

- Some optional applications, such as MirrorView™ and SnapView™, require private LUNs. These LUNs are no longer treated special in the cache as they were in VNX OE 5.31, but work well with the new MCx architecture. Therefore, EMC recommends that you disable Multicore FAST Cache on MirrorView’s Write Intent Log and SnapView’s Clone Private LUNs to prevent unnecessary promotions into the Multicore FAST Cache.

- SnapView snapshots and related replication software, such as MirrorView/A and SAN Copy™ (incremental sessions), require Reserved LUNs. Multicore FAST Cache does not improve Reserved LUN performance, but it is not a detriment to performance as with Write Intent Logs and Clone Private LUNs. Disabling Multicore FAST Cache for reserved LUNs can help to minimize the overall Multicore FAST Cache workload. Multicore FAST Cache can be disabled at the LUN level if the Reserved LUN is created in RAID groups. If the Reserved LUNs are created in a pool and there are other LUNs in the pool that need Multicore FAST Cache, they can be left with Multicore FAST Cache enabled.
- Multicore FAST Cache consumes a portion of the storage system memory that was formerly available for Multicore Cache. The amount of memory consumed is dependent on the storage system model and Multicore FAST Cache size.
- Flash drives installed in vault drive locations cannot be used to create Multicore FAST Cache. The VNX2 OE restricts this usage.
- Before enabling Data At Rest Encryption (D@RE) on the system, Multicore FAST Cache must be fully disabled. Upon disabling Multicore FAST Cache, the entire contents of Multicore FAST Cache will be flushed to disk. After D@RE is enabled, Multicore FAST Cache can be re-enabled. This will require a re-warm up of the data.

Failure Handling

VNX2 global hot sparing algorithms are used for the Flash drives configured as Multicore FAST Cache. Global hot sparing provides automatic, online rebuilds of redundant RAID groups when any of the group's drives fail. EMC has further advanced this functionality with proactive hot sparing. Proactive hot sparing recognizes when a drive is nearing failure and preemptively copies the drive content before it fails. The combination of these features minimizes each RAID group's vulnerability to additional drive failures and prevents data loss. For performance reasons, only FAST Cache optimized drives replace failing Flash drives in Multicore FAST Cache.

If a single disk failure within a Multicore FAST Cache group occurs, the underlying mirrored pair (RAID 1) of drives to which the failed disk belongs enters a degraded mode. The cache page cleaning algorithm then increases the rate at which Multicore FAST Cache pages are copied from Multicore FAST Cache drives to HDDs for the degraded group, and all dirty cache pages in the group are copied to the back end drives. Only read operations are allowed to this Multicore FAST Cache group to reduce the potential exposure to data loss of a drive failure in a non-redundant RAID group.

While in this state, write operations that were destined for the degraded group are now serviced by the remaining groups. Once the degraded group is repaired, either by a permanent hot spare or by drive replacement, the degraded group exits degraded mode and seamlessly starts to re-warm data into it.

Note: FAST VP Optimized drives cannot be used as spares for FAST Cache Optimized drives.

Conclusion

Multicore FAST Cache allows the storage system to provide Flash-drive class performance to data with a high locality of reference. This working data set increases IOPs without placing all of the data onto Flash drives. Multicore FAST Cache absorbs I/O bursts from applications, thereby reducing the load on HDDs, which helps to improve the TCO of the storage solution. You can manage Multicore FAST Cache through Unisphere in an easy, intuitive manner.

Multicore FAST Cache works in a complementary way with FAST VP technology. Both technologies help ensure data is placed on the most appropriate storage tier based on their usage pattern.

References

The following white papers are available on EMC Online Support:

- *EMC VNX Unified Best Practices For Performance*
- *EMC Unified Storage System Fundamentals for Performance and Availability*
- *EMC Unisphere: Unified Storage Management Solution For The VNX2 Series*
- *EMC VNX2 FAST VP*
- *Virtual Provisioning For The VNX2 Series*
- *EMC VNX2: Data-At-Rest Encryption*
- *EMC® Infrastructure for VMware® View™ 5.0*
- *Leveraging EMC FAST Cache with Oracle OLTP Database Applications*

Appendix A Multicore FAST Cache Configuration Options

Table 2. Multicore FAST Cache maximum configuration options with SAS FLASH drives

Storage System	Flash (SSD) Disk Capacity	Maximum Multicore FAST Cache Capacity
VNX5200	100 GB / 200 GB	300 GB (6 drives) / 600 GB (6 drives)
VNX5400	100 GB / 200 GB	500 GB (10 drives) / 1 TB (10 drives)
VNX5600	100 GB / 200 GB	1 TB (20 drives) / 2 TB (20 drives)
VNX5800	100 GB / 200 GB	1.5 TB (30 drives) / 3 TB (30 drives)
VNX7600	100 GB / 200 GB	2.1 TB (42 drives) / 4.2 TB (42 drives)
VNX8000	100 GB / 200 GB	2.1 TB (42 drives) / 4.8 TB (48 drives)

Table 3. Multicore FAST Cache maximum configuration options with SAS FLASH 2 drives

Storage System	Flash (SSD) Disk Capacity	Maximum Multicore FAST Cache Capacity
VNX5200	200 GB	600 GB (6 drives)
VNX5400	200 GB / 400 GB	1 TB (10 drives) / 800 GB (4 drives)
VNX5600	200 GB / 400 GB	1 TB (10 drives) / 2 TB (10 drives)
VNX5800	200 GB / 400 GB	1.4 TB (14 drives) / 2.8 TB (14 drives)
VNX7600	200 GB / 400 GB	2 TB (20 drives) / 4 TB (20 drives)
VNX8000	200 GB / 400 GB	2 TB (20 drives) / 4.8 TB (24 drives)

Note: EMC suggests utilizing SAS Flash drives in Multicore FAST Cache for configurations of 600 GBs and less.

Appendix B FAST VP and Multicore FAST Cache

Fully Automated Storage Tiering for Virtual Pools (FAST VP) is a feature that performs storage tiering for 256 MB slices of data at a sub-LUN level in pools that contain multiple drive types. FAST VP automatically moves more *active* slices (data that is more frequently accessed) to the best performing storage tier, and it moves less active slices to a lower performing (and less expensive) tier for a better TCO. For more details on this feature, refer to the *EMC VNX FAST VP* white paper available on EMC Online Support.

Table 4. Comparison between the FAST VP and Multicore FAST Cache features

Multicore FAST™ Cache	FAST™ VP
Allows FAST Cache optimized Flash drives to be used to extend the existing caching capacity of the storage system.	Allows a single LUN to leverage the advantages of multiple drive types through the use of storage pools.
Granularity is 64 KB.	Granularity is 256 MB.
Data that is accessed frequently is <i>copied</i> from HDDs to FAST Cache Optimized Flash drives.	Data is <i>moved</i> between different storage tiers based on weighted-average-of-access statistics collected over a period of time.
Use when workload changes are unpredictable and very dynamic, and require a quick response time.	Use when workload pattern changes are predictable and relatively low.
Constantly promotes frequently accessed HDD data to Multicore FAST Cache. There are no relocation cycles.	Data movement occurs in scheduled or manually invoked relocation windows.
Real-time monitoring decides which data needs to be promoted to Multicore FAST Cache.	Hourly analysis decides which portion of data needs to be moved.

You can use Multicore FAST Cache and FAST VP together to yield high performance and TCO from the storage system. For example, you can use FAST Cache Optimized Flash drives to create Multicore FAST Cache, and use FAST VP for storage pools consisting of SAS and NL-SAS disk drives. From a performance point of view, Multicore FAST Cache provides an immediate performance benefit to bursty data, while FAST VP moves more active data to SAS drives and less active data to NL-SAS drives. From a TCO perspective, Multicore FAST Cache can service active data with fewer Flash drives, while FAST VP optimizes disk utilization and efficiency with SAS and NL-SAS drives.

As a general rule, use Multicore FAST Cache in cases where storage system performance must be improved immediately for burst-prone data with a high locality of reference. On the other hand, FAST VP optimizes TCO by moving data to the appropriate storage tier based on sustained data access and demands over time. Multicore FAST Cache focuses on improving performance while FAST VP focuses on improving TCO. These features complement each other and, when used together, can improve performance and TCO.

Multicore FAST Cache works with FAST VP to ensure that resources are not wasted on unnecessary tasks. For example:

- If FAST VP moves a slice of data to Flash drives, Multicore FAST Cache will not promote a 64KB chunk of data from that slice into Multicore FAST Cache, even if the criteria is met for promotion. This ensures that resources are not wasted by copying data from one Flash drive to another.
- If a bursty workload starts accessing a particular 64KB chunk within a slice of a Multicore FAST Cache-enabled LUN, FAST VP does not immediately move that slice to a different storage tier. Instead, Multicore FAST Cache promotes the 64KB chunk into the cache. After the 64KB chunk is promoted, a majority of I/O operations will be serviced from Multicore FAST Cache. This can result in less activity on the back-end LUNs, and FAST VP might not need to move the slice to a higher-storage tier. In this case, a FAST VP-initiated data move is avoided when there is a temporary burst in an application's workload.
- In contrast to the previous scenario, if the application workload has increased on a sustained basis, Multicore FAST Cache will need to write data back into the HDD LUNs to make space for new promotions. This will register as back-end activity and FAST VP will eventually schedule a move of the data slices to higher storage tier—which may be Flash drives. When this move is completed, Multicore FAST Cache does not promote any data that is already in the Flash drive storage tier.
- You will see higher performance benefits and faster reaction time for changing I/O usage patterns when using FAST Cache Optimized Flash drives for Multicore FAST Cache. The downside of higher parity overhead in Multicore FAST Cache because of the RAID 1 architecture is offset by improved performance of the DRAM cache. Performance gains are due to the flushes from DRAM of an I/O that is in Multicore FAST Cache to the Flash drives instead of back to HDDs.

Appendix C Multicore FAST Cache Comparison with Storage System Cache

Multicore FAST Cache is semi-conductor-based storage technology. It provides a large-capacity secondary tier of Flash-memory-based caching between the storage system's fast, limited-capacity DRAM cache, and slower, higher-capacity HDDs.

Table 5. Comparison of DRAM memory and Multicore FAST Cache

Characteristic	DRAM Cache	Multicore FAST Cache
Position	Closest to the CPU, and has the lowest latency.	A step further from the CPU and is slower than DRAM cache.
Response time	Response time is from nanoseconds to microseconds.	Response time is from microseconds to milliseconds.
Upgradeability	Not upgradeable.	Upgradeable in all supported models, and the options depend on storage system model and size of Flash drives.
Operation	Has a single area that serves read and write operations.	Upgradeable in all supported models, and the options depend on storage system model and size of Flash drives.
Capacity	Is limited in size, compared to Multicore FAST Cache.	Can scale to much larger capacities.
Granularity	Has very high granularity, which is effectively the I/O size. Each cache page is fixed at a size of 8 KB.	Operates in extents of 64 KB granularity.
Availability	In case of failure, replacement requires service by qualified personnel.	In case of failure, another Flash-drive hot spare automatically replaces the failing drive, and the faulted component is customer-replaceable.
Power failure	Contents are volatile and cannot withstand a power loss.	Contents are non-volatile and can withstand a power loss.