Introduction to the EMC VNXe3200 FAST™ Suite

Overview

Abstract
This white paper is an introduction to the EMC® FAST™ Suite for the VNXe3200 storage system. It describes the components of the EMC FAST Suite: The FAST Cache and FAST VP, as well as their implementation in the VNXe3200. Usage guidelines and major customer benefits are also included.

February 2015
# Table of Contents

## Executive summary

- Audience ................................................................. 4
- Terminology ............................................................ 4

## Multicore FAST Cache ................................................. 7

- Introduction ............................................................. 7
- Multicore FAST Cache components .......................... 7
  - Policy Engine ..................................................... 7
  - Memory Map ..................................................... 7
- Theory of operation ................................................. 7
  - Multicore FAST Cache promotions ........................ 7
  - Host read operations ........................................... 8
  - Host write operations .......................................... 9
  - Multicore FAST Cache page cleaning ..................... 10
  - Configuring FAST Cache with Unisphere .............. 11
  - Drive configuration ........................................... 13
  - Failure handling ................................................. 14

## FAST VP .................................................................... 15

- Introduction ............................................................. 15
  - FAST VP licensing ................................................. 15
  - Using FAST VP ..................................................... 16
  - Storage pools ..................................................... 16
  - Storage tiers ..................................................... 17
  - FAST VP management ......................................... 20
  - Tiering policies ................................................... 21
  - Managing FAST VP at the storage pool level........... 23
  - Relocation schedule ............................................. 25
  - Manual relocation ................................................ 28
  - FAST VP operations ............................................ 28
  - FAST VP algorithm ............................................. 29

## Comparison and Usage of Multicore FAST Cache and FAST VP ........... 32

- Interoperability ........................................................ 33
  - Multicore Cache ................................................... 34
  - File-Level Retention (FLR) .................................... 35
  - File deduplication and compression ....................... 35
  - Thin storage resources ......................................... 35

## Conclusion ................................................................ 35

## References ................................................................ 36
Executive summary

Since the original deployment of flash technology in enterprise arrays, EMC® has expanded the use of this technology throughout its midrange offerings. The VNXe FAST™ Suite software allows the new VNXe3200 to fully leverage the higher performance of flash disks. The FAST Suite consists of two software components: Multicore FAST Cache (FAST Cache) and Fully Automated Storage Tiering for Virtual Pools (FAST VP). These features work together to ensure that the system’s flash storage is used as efficiently as possible and the most active data is serviced from these higher performance disks.

FAST Cache allows the creation of a high-capacity secondary cache that sits between the Storage Processor’s DRAM cache and hard disk drives (HDDs). FAST Cache supplements the system’s existing cache by using system flash disks. Frequently accessed data residing on HDD’s is copied to the FAST Cache, resulting in improved access times and higher system performance. Because it promotes data from rotating media to solid state drives in 64K chunks FAST Cache is by far the most cost efficient way to provision with flash disks. It is quite possible that the active data set in a small capacity system will fit within FAST Cache.

When FAST VP is enabled, the storage system measures, analyzes, and implements a dynamic storage-tiering policy in a faster and more efficient way than a human analyst. FAST VP intelligently manages data placement based on historical access, moving “hotter” data up to the higher performing (flash or SAS) disks and “colder” data down to lower performing (SAS or NL-SAS) disks. The FAST Suite makes efficient use of flash capacity. Data which resides in tiers on flash disks will not be promoted to FAST Cache preserving that resource as a performance buffer.

FAST Cache and FAST VP are licensed features available on the VNXe3200. These licenses are available as part of the FAST Suite for customers who want to make the most efficient use of flash disks in their arrays.

Audience

This white paper is intended for EMC customers, partners, and employees who are considering the use of the FAST Cache and FAST VP features in the VNXe3200 storage system. It assumes familiarity with VNXe storage systems and EMC’s management software.

Terminology

**Cache page** - Unit of allocation inside the FAST Cache, 64 KB in size.

**Cache clean page** - Page of the FAST Cache that is valid and contains a copy of data that has been synchronized with the storage resource.

**Cache dirty page** - Page of the FAST Cache that is valid and contains the most recent copy of data, but has not yet been synchronized with the storage resource.

**Chunk** - Portion of data in a particular address range (64 KB).

**DRAM memory** - Storage-system component used by the Multicore Cache to store data in very fast storage media (DRAM), to serve requests for that data faster.
**FAST Cache SSD** – The only flash disks that may be used to create the FAST Cache. These disks use single-level cell (SLC) technology, which provides better write performance and write endurance than enterprise multi-level cell (eMLC) flash disks. FAST Cache SSDs can also be used for storage pool provisioning.

**FAST VP SSD** – Flash disks recommended for use in pools and with FAST VP. These flash disks use enterprise multi-level cell (eMLC) technology. These disks are used when increased write endurance is not necessary, or affordability is a higher priority. FAST VP SSDs cannot be used to create the FAST Cache.

**Flash disk (SSD)** - Also known as a solid-state drive, this is a data storage device that uses solid-state media to store data. Because it does not have moving parts, a flash disk provides extremely low response times and high IOPS compared to rotating hard-disk drives (HDDs).

**Hard disk drive (HDD)** - Data storage device that stores data on magnetic surfaces and rotates at various speeds.

**Locality of reference** - Concept that logical blocks located close to each other are accessed repeatedly at approximately the same time.

**Logical block address (LBA)** - Offset into a particular LUN to provide a way of locating particular data.

**LUN** - Logical unit of storage created in a pool. A LUN can be either a thin LUN or a thick LUN.

**Memory map** - Container of addresses in which each entry represents a FAST Cache page. This map shows which pages are in the FAST Cache and where they are located. A copy of the memory map resides in DRAM memory. This ensures that the pages are accessed at memory speeds. A copy of the memory map is also stored on the flash disks, which ensures non-volatility.

**Multicore Cache (DRAM Cache)** - Software component that improves host write and read performance by using the VNXe storage processors’ DRAM.

**Multicore Cache Hit** - Instance when an IO can be completed using the contents of the Multicore Cache.

**Multicore Cache Miss** - Instance when the Multicore Cache does not contain the data needed to complete the IO.

**Multicore FAST Cache copy** - Process of copying data from a FAST Cache page to the original user LUN.

**Multicore FAST Cache promotion** - Process of copying data from a storage resource to a FAST Cache page.

**Multicore FAST Cache flush** - Process of copying data from a FAST Cache page to a storage resource, freeing the page for use.

**Multicore FAST Cache hit** - Instance when an IO can be completed using the contents of the FAST Cache.
**Multicore FAST Cache miss** - Instance when the FAST Cache does not contain the data needed to complete the IO, and the HDD must be used.

**Rebalance** - Automated data relocation to correct the data imbalance created when new, empty disks are added to an existing pool.

**Redundant Array of Independent Disks (RAID)** - Data storage technology where data is stored on multiple disk drives to increase performance and provide redundancy and fault tolerance.

**Skew** – When a small percentage of capacity is responsible for most of the IO activity.

**Slice** – Incremental unit of capacity that is allocated to a pool LUN. 256 MB in size.

**Slice Relocation** - Movement of 256MB blocks of data within or between tiers, based on the relative activity level (temperature) of each block.

**Storage Pool** - Group of disk drives used for configuring LUNs, VMware datastores, and file systems (thick and thin). Disks can only be a member of one pool.

**Storage Resource** – A LUN, VMware datastores, or file system. It also refers specifically to the copy of the data that is stored on the storage pool drives, and not the copy in DRAM Cache or FAST Cache.

**Temperature** – Weighted moving average of a slice’s activity level over time.

**Thick Storage Resource** - Type of pool storage resource in which physical space is allocated upon creation and is equal to the user capacity seen by the host server.

**Thin Storage Resource** - Type of pool storage resource where physical space is allocated on-demand and can be less than the user capacity seen by the host server.

**Tier** - Different categories of media in a storage pool. These categories provide various levels of performance and capacity through several disk types. Available tiers are Extreme Performance (flash), Performance (SAS), and Capacity (NL-SAS).
**Multicore FAST Cache**

**Introduction**

A component of the FAST Suite, the FAST Cache leverages the performance of flash disks and efficiently uses them to store the most accessed data within the system. If a particular chunk of data residing on hard disks (HDD's) is accessed frequently, the VNXe3200 automatically copies that chunk from the hard disks to the flash disks used by the FAST Cache. Subsequent IO access to the same chunk is serviced at flash disk 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 the FAST Cache, the least recently used data is written to HDD and replaced first.

**Multicore FAST Cache components**

The FAST Cache requires the FAST Cache license to take advantage of the feature. To create the FAST Cache, you must have at minimum two FAST Cache SSDs in the system, configured as a RAID 1 mirrored pair. As discussed in the previous *Terminology* section, only FAST Cache SSDs may be used to create the FAST Cache. For differences between FAST Cache SSDs and FAST VP SSDs, refer to the *Terminology* section. Once the license is installed, the system uses the following main components to process and execute the FAST Cache:

**Policy Engine**

The Policy Engine monitors the flow of IO through the FAST Cache and determines what data gets copied to the FAST Cache. When a chunk of data on a LUN in a FAST-Cache-enabled pool is accessed three times in a short period of time, the chunk is copied to the FAST Cache. There are a few exceptions to this general rule that will be discussed later. The Policy Engine also maintains statistical information about the data access.

**Memory Map**

The Memory Map tracks the location of all 64 KB chunks in the FAST Cache. It resides in DRAM memory to ensure the best performance when locating data. A copy of the Memory Map is also maintained on the drives in the FAST Cache for high availability. After querying the Memory Map to determine whether a chunk is in FAST Cache, the system can direct IO for that chunk accordingly.

**Theory of operation**

**Multicore FAST Cache promotions**

A FAST Cache promotion is the process by which data is copied from spinning media HDDs and placed into the 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. After being promoted, a copy of the data continues to reside on the spinning media.
A promotion occurs when the Policy Engine determines that a 64 KB chunk is being accessed frequently and that performance would be improved by copying the chunk to the FAST Cache. A chunk is promoted after it is accessed three times within a certain period of time, which is determined dynamically based on the IO workload. The Memory Map is then updated to indicate that the promoted chunk is now in the FAST Cache. The next time this data is accessed (assuming a DRAM Cache miss) the Memory Map is checked, and the map shows that the data is in the FAST Cache. Then the IO is serviced from the FAST Cache instead of spinning media. Because the data is now accessed from flash disks, applications will experience lower response times and higher IOPS. If a substantial part of the working set is copied to the FAST Cache over time, applications can see higher average performance, even with lower performing HDDs in the pool.

There are several circumstances in which copying a chunk into the FAST Cache does not improve performance, or where the IO is better handled by the DRAM Cache:

- The data already resides on flash disks in the pool
  A promotion to FAST Cache will not occur
- Small block sequential IO
  The DRAM Cache has built-in optimization for this type of workload, such as prefetching and coalescing.
- High-frequency access patterns
  Data accessed very frequently will continue to reside in the DRAM Cache and will not be flushed frequently enough via the Least Recently Used algorithm to result in repeated storage resource access.
- Zero fill requests
  Re-writing a region with zeros. Since this is a one-time operation and not repeated access, this will not benefit from the FAST Cache.
- IOs larger than 64 KB
  In these cases, promotions do not occur, even if the chunk is accessed frequently.

**Host read operations**

Incoming read IO from a host application is first checked against the content of the Multicore Cache (DRAM Cache). If the IO can be serviced from the DRAM Cache, the IO is completed. The FAST Cache Memory Map is not accessed during this exchange.

If the data is not in the DRAM Cache, a DRAM Cache miss occurs, and the system checks the FAST Cache Memory Map to determine whether the IO is for a chunk that already exists in the FAST Cache. If the data exists in the FAST Cache, the Policy Engine redirects the IO request to the FAST Cache. The data is then copied from the FAST Cache to the DRAM Cache, which then satisfies the read.

If the data is not in the FAST Cache, the requested data is copied from the HDDs into the DRAM Cache. The DRAM Cache satisfies the read request at this time. If this operation
occurs three times within a certain time frame, the chunks are promoted to the FAST Cache. Re-hits while the chunk resides in the DRAM Cache do not count toward a FAST Cache promotion. Figure 1 shows the read operation.

![Multicore FAST Cache Read operation](image)

**Figure 1. Multicore FAST Cache Read operation**

**Host write operations**

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

The DRAM Cache later copies the data to the FAST Cache or the storage resource as part of its cleaning process, shown in Figure 2 below. After being copied to the user LUN from the DRAM Cache, the Policy Engine may determine that the data should be promoted to FAST Cache, in which case it is copied there directly from the DRAM Cache. This results in a clean DRAM Cache and FAST Cache pages, as the same data now resides in all three locations.

![Multicore Cache copy operation](image)

**Figure 2. Multicore Cache copy operation**
In the instance where write caching is disabled on the system due to hardware faults and the FAST Cache is enabled, the DRAM Cache temporarily holds the IO and checks the Memory Map to see if the data exists in the FAST Cache. If the data exists in the FAST Cache, the DRAM Cache updates the data in the FAST Cache and Memory Map, and then completes the IO by sending an acknowledgment to the host. If the data is not contained within the FAST Cache, the DRAM Cache updates the data on the hard disk drives. For frequently used data, the Policy Engine copies the data into the FAST Cache as a clean cache page. Figure 3 further depicts this scenario.

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

**Figure 3. Multicore Cache write-through operation (Write Cache Disabled)**

**Multicore FAST Cache page cleaning**

FAST Cache pages are 64 KB units that correspond to 64 KB chunks that have been promoted into the FAST Cache. Because the FAST Cache can be written to, pages containing data can either be dirty or clean. Dirty pages contain data that was modified in the FAST Cache but not yet updated on the storage resource. When a dirty page’s contents are copied to the corresponding location on the storage resource, the data in the FAST Cache matches the data on the storage resource. At this point, the page is considered clean. The FAST Cache contains a cleaning process that proactively copies dirty pages to the underlying physical devices during times of minimal activity. When this happens, the page that was copied becomes clean and remains in the FAST Cache in case its data is requested again. The FAST Cache chooses pages to use for promotions in the following order:

1. Use free unused pages.
2. Overwrite the least recently used clean pages.
3. Flush the least recently used dirty pages to make them clean, and then overwrite these pages.
Promoting to free and clean pages is faster than promoting to dirty pages, because dirty pages must be copied to the storage resource (cleaned) before new data can be promoted. This greater speed when promoting data to the FAST Cache is the reason for the proactive cleaning process explained above. If there are no free or clean pages available for a new promotion, the FAST Cache flushes dirty pages to the storage resource before using these pages for promotions.

**Configuring FAST Cache with Unisphere**

To configure the FAST Cache in Unisphere, navigate to **Storage > Storage Configuration > FAST Cache**.

If the FAST Cache has not been created on the storage system and FAST Cache SSDs are available, the **Create** button in the bottom of the FAST Cache page is available. Click **Create** to launch the FAST Cache Wizard (Figure 4).

![Figure 4. FAST Cache Wizard](image)

The wizard displays the usable capacity of available flash disks that can be used to create the FAST Cache. In the VNXe3200, the FAST Cache is made up of flash disks in a RAID 1 mirrored pair. You can specify whether existing pools should automatically use the newly configured FAST Cache. Existing pools use the FAST Cache by default unless this is changed.
here. You can modify this setting later at the pool level if desired. Click Next to verify your selections and then Finish to create the FAST Cache.

After you create the FAST Cache, the Unisphere FAST Cache page shows the status, total space, and number of disks in the General tab, as well as which disks comprise the FAST Cache in the Disks tab.

Although the FAST Cache is a global resource, it is enabled on a per pool basis. All LUNs created in the storage pool will have FAST Cache enabled (default) or disabled collectively. You can configure a pool to use the FAST Cache during pool creation in Step 4 of the wizard, as shown in Figure 5.

Figure 5. Configure FAST Cache usage during pool creation
For existing pools, use the **Settings** tab in the Storage Pool details page to enable the FAST Cache (Figure 6). Check **Allow this pool to use the FAST Cache** and click **Apply**.

![Settings tab on the storage pool details page](image)

**Drive configuration**

During the FAST Cache creation, you can specify the capacity by the number of drives used. Once the FAST Cache configuration is set, it cannot be changed. Table 2. RAID Configuration Options shows the drive quantities, sizes, and capacities that are supported for the FAST Cache configuration.

<table>
<thead>
<tr>
<th>Drive Quantity</th>
<th>Drive Size (GB)</th>
<th>FAST Cache Capacity (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>300</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>400</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>200</td>
<td>400</td>
</tr>
</tbody>
</table>

**Table 1. Drive Configuration Options**
**Failure handling**

If a single drive failure within Multicore FAST Cache group occurs, the underlying mirrored pair (RAID 1) of drives to which the failed drive 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 drive 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 in the situation the remaining drive fails.

While in this state, write operations that were targeted 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.

Although the disks used for the FAST Cache can automatically and permanently spare like other VNXe disks, the Hot Spare Policy does not apply to disks in the FAST Cache. This means that at minimum only two FAST Cache SSDs are required to create the FAST Cache. However, it is recommended to have a spare disk available in the case of failure. If the array has additional FAST Cache SSDs not being used for the FAST Cache, these disks are subject to the Hot Spare Policy of 1 spare per 30 disks.
FAST VP
Introduction
While the FAST Cache is excellent for data that is the most frequently accessed at the moment, data also has a longer term lifecycle that must be considered. As data progresses through its lifecycle, it experiences varying levels of activity. When data is first created, it is typically heavily accessed. As data ages, it is typically accessed less often. Fully Automated Storage Tiering for Virtual Pools (FAST VP) is a simple solution for automatically and dynamically matching storage requirements with changes in the frequency of data access. FAST VP segregates disk drives into three categories, called tiers:

- **Extreme Performance Tier** – Flash disks
- **Performance Tier** – Serial Attached SCSI (SAS) disks
- **Capacity Tier** – Near-Line SAS (NL-SAS) disks

FAST VP is a feature designed to aggressively reduce Total Cost of Ownership (TCO) by increasing storage efficiency while maintaining performance. Rather than using only one type of disk in a pool, mixing flash, SAS, and NL-SAS disks allows you to maintain performance requirements while reducing the disk count by leveraging high-capacity NL-SAS. This in turn reduces the total storage cost. In some cases, a nearly two-thirds reduction in disk count can be achieved, while in other cases, performance throughput can double by simply adding an average of 5 percent of a pool’s total capacity in flash disks.

The VNXe3200 delivers high value by providing a unified approach to auto-tiering for file and block data. Both block and file data can use virtual pools and FAST VP. This provides compelling value for users who want to optimize the use of high-performance disks across their environment while still getting the benefit of low-cost capacity drives for TCO.

FAST VP is designed to complement FAST Cache when they are used together. FAST VP helps to gain TCO benefits and improve efficiency while FAST Cache boosts overall system performance.

**FAST VP licensing**
When a FAST VP license is installed, users have access to all the features and management options described in the FAST VP sections of this white paper. The following features are enabled when a FAST VP license is installed:

- Ability to create a multi-tier pool
- Ability to set LUN tiering policies
- Ability to access the Data Efficiency Settings page in Unisphere
- Ability to access the FAST VP tabs on storage pools or storage resources

If a user installs the FAST VP license after creating storage resources, the tiering policies of those resources will be set to “Start High then Auto-Tier” by default. Rebalancing when expanding a single-tier pool (discussed later) will occur regardless of whether FAST VP is
licensed on the array. The features discussed in the following sections of this white paper assume that the array is licensed for FAST VP.

Using FAST VP

Storage pools

Storage pools provide a framework that allows FAST VP to take full advantage of different disk types. A pool is a physical collection of disks on which logical units (LUNs), VMware datastores, and file systems are created. Pools can contain a few disks or hundreds of disks. Because of the large number of disks supported in a pool, pool-based provisioning spreads workloads over more resources thus minimizing planning and management efforts.

Pools can be homogeneous - containing a single disk type (flash, SAS, or NL-SAS), or heterogeneous - containing multiple disk types.

Homogeneous pools

Homogeneous pools are recommended for applications with limited skew, such that their access profiles can be very random across a large address range. Multiple LUNs with similar profiles can share the same pool resources. These LUNs provide more predictable performance based on the disk type employed. In a homogeneous pool, only one disk type (flash, SAS, or NL-SAS) is selected during pool creation.

![Image of homogeneous storage pools]

**Figure 7. Homogeneous storage pools**

As seen in Figure 7, each pool contains various sized LUNs that consist of a single disk type.
**Heterogeneous pools**

Heterogeneous pools consist of multiple disk types. The VNXe3200 supports flash, SAS, and NL-SAS disks in the same pool. As shown in Figure 8, there can be a maximum of three disk types in a heterogeneous pool. Data in a particular LUN can reside on some or all of the different disk types.

![Figure 8. Heterogeneous storage pools](image)

**Storage tiers**

When creating storage pools on a VNXe3200, data can use three different categories of media in a single pool. These categories, referred to as storage tiers, provide various levels of performance and capacity through several disk types. Available tiers for creating storage pools include:

- **Extreme Performance Tier** – Flash disks
- **Performance Tier** – Serial Attached SCSI (SAS) disks
- **Capacity Tier** – Near-Line SAS (NL-SAS) disks

FAST VP differentiates each of these tiers by disk type, but it does not differentiate on the basis of rotational speed. EMC strongly encourages that you avoid mixing rotational speeds per disk type in a given pool. If multiple rotational-speed SAS disks exist in the array, you should configure them in separate pools from each other.

FAST VP can leverage one, two, or all three storage tiers in a single pool. Each tier offers unique advantages in terms of performance and cost.
Per-tier RAID configuration

During storage pool creation, you can select RAID protection on a per-tier basis. Each tier can only have a single RAID type, and once the RAID configuration is set for that tier in the pool, it cannot be changed. Table 2. RAID Configuration Options shows the RAID configurations that are supported for each tier.

<table>
<thead>
<tr>
<th>RAID Type</th>
<th>Default Configuration</th>
<th>Supported Configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAID 1/0</td>
<td>4+4</td>
<td>1+1*, 2+2, 3+3, 4+4</td>
</tr>
<tr>
<td>RAID 5</td>
<td>4+1</td>
<td>4+1, 8+1, 12+1</td>
</tr>
<tr>
<td>RAID 6</td>
<td>6+2</td>
<td>6+2, 8+2, 10+2, 14+2</td>
</tr>
</tbody>
</table>

Table 2. RAID Configuration Options

*RAID 1/0 (1+1) is RAID 1

Configurations with higher data to parity ratios (12+1, 14+2, etc.) provide capacity savings compared to those with lower data-to-parity ratios. The tradeoffs for higher data to parity ratios are larger fault domains and potentially longer rebuild times. This is especially true for RAID 5, which has only a single parity disk. EMC advises you to choose carefully between RAID 5 configurations and decide whether robustness or efficiency is a higher priority. Robustness is less likely to be an issue for RAID 6, because it has two parity disks.

When creating a storage pool with any of the available tiers, you must add disks in multiples of the supported RAID configurations.

Extreme Performance tier

The Extreme Performance tier is used when response times and performance are the most important criteria for storage. The Extreme Performance tier uses flash technology – solid-state drives (SSDs) that contain no moving parts. This technology eliminates rotational latencies and can lead to a significant performance boost and significant energy savings.

Tests show that adding a small percentage of flash capacity to storage and using intelligent tiering products, such as FAST VP, can deliver double-digit percentage gains in throughput and response time performance in some applications.

Flash disks have a higher cost per GB but a lower cost per IO compared to traditional spinning disks. To receive the best return, use flash disks for data that requires fast response times and high IOPS. FAST VP enables you to optimize the use of these high-performance resources, because it automatically relocates “hot” data to the flash disks at a sub-LUN level.

Performance tier

The Performance tier achieves a combination of performance and capacity. This tier, composed of Serial Attached SCSI (SAS) disks, offers high levels of performance, reliability, and capacity. SAS disks are based on industry-standardized, enterprise-level, mechanical hard-drive technology that stores digital data on a series of rapidly rotating magnetic platters.
The Performance tier includes both 10K and 15K RPM spinning disks, which are available on all EMC midrange storage systems. When adding these disks to a storage pool, EMC recommends using a single rotational speed per pool. The Performance tier is valuable, because it offers high, all-around performance with consistent response times, high throughput, and good bandwidth at a mid-level price point.

**Capacity tier**

The Capacity tier is used to decrease the cost per GB of data. This tier, consisting of 7.2K RPM Near-Line SAS (NL-SAS) disks, is designed for maximum capacity at a modest performance level. Although NL-SAS disks have a slower rotational speed compared to disks in the Performance tier, NL-SAS disks can significantly reduce energy use and free up capacity in the more expensive and higher performing storage tiers.

In a typical system, 75-95% of application data has little IO activity. Since NL-SAS disks cost less than performance disks on a per-GB basis, and their cost is a small fraction of the cost of flash disks, they are the most appropriate type of media for this “cold” data. NL-SAS disks consume 96% less power per terabyte than performance disks, and offer a compelling opportunity for TCO improvement that considers both purchase cost and operational efficiency.
Considerations for using tiers

When all three tiers are available, there are considerations for using one tier or another. For example, response times are lower with flash disks compared to NL-SAS disks, but capacity is greater with NL-SAS disks than with flash disks. Table 3. Comparison of the Extreme Performance, Performance, & Capacity Tiers compares the three tiers.

<table>
<thead>
<tr>
<th></th>
<th>Extreme Performance (Flash)</th>
<th>Performance (SAS)</th>
<th>Capacity (NL-SAS)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Response Time</strong></td>
<td>1-5 ms</td>
<td>≈5 ms</td>
<td>7-10 ms</td>
</tr>
<tr>
<td><strong>Multi-Access Response Time</strong></td>
<td>&gt; 10 ms</td>
<td>10-50 ms</td>
<td>≤ 100ms</td>
</tr>
<tr>
<td>High IOPS/GB and Low Latency</td>
<td>High bandwidth with contending workloads</td>
<td>Low IOPS/GB</td>
<td>Leverages storage array SP cache for sequential and large block access</td>
</tr>
<tr>
<td>Provides extremely fast access for reads</td>
<td>Sequential reads leverage read-ahead</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Executes multiple sequential streams better than SAS</td>
<td>Sequential writes leverage system optimizations favoring disks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Writes slower than reads</td>
<td>Read/write mixes provide predictable performance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heavy concurrent writes affect read rates</td>
<td>Uncached writes are slower than reads</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single-threaded, large, sequential IO is equivalent to SAS</td>
<td>Long response times for heavy-write loads</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Comparison of the Extreme Performance, Performance, & Capacity Tiers

**FAST VP management**

Although FAST VP is a feature that is implemented for the entire storage pool, you can modify settings at the storage resource level. During the creation of a storage resource via the GUI or CLI, you can define a tiering policy. A tiering policy specifies where the initial placement of data will be and how that data will be relocated within the pool during scheduled and manually invoked relocation periods. FAST VP bases decisions for how data relocation occurs on performance statistics collected every hour. For more information, see the “FAST VP Algorithm” section in this paper.
Tiering policies

FAST VP is a completely automated feature that implements a set of user-defined tiering policies to ensure the best performance for various business needs. The relocations occur for each 256 MB set of logically ordered blocks, called a slice. FAST VP uses an algorithm to make data relocation decisions based on the activity level of each slice. It ranks the order of data relocation across all storage resources within each separate pool. The system uses this information in combination with the per-storage resource tiering policy to create a candidate list for data movement.

The following storage resource level policies are available, and are described in detail below:

- **Highest Available Tier**
- **Auto-Tier**
- **Start High then Auto-Tier (Recommended)**
- **Lowest Available Tier**
- **No Data Movement**

Users can set any storage resource level policy except the “No Data Movement” policy both during and after storage resource creation. The “No Data Movement” policy is only available after storage resource creation.

**Highest Available Tier**

Use the “Highest Available Tier” policy when quick response times are a priority. This tier is effective for storage resources which, although not always the most active, require highest levels of performance whenever they are accessed. The “Highest Available Tier” policy initially places data into the highest available tier. If this tier does not have enough available space for the entire storage resource, the remaining slices are placed into the next highest tier with free capacity.

The “Highest Available Tier” may not always be the highest tier. Slices are prioritized for the highest tier as follows:

- Existing slices currently residing in the highest tier have priority at the time a new LUN is created. New slices will not immediately force existing slices out of the highest tier, regardless of tiering policies, until a FAST VP relocation occurs.
- Slices of all LUNs with the “Highest Available Tier” policy then compete for the highest tier based on their temperature, but always have priority over slices of storage resources with other tiering policies.
Auto-Tier

A small portion of a large set of data may be responsible for most of the IO activity in a system. FAST VP allows for moving a small percentage of the “hot” data to higher tiers while maintaining the rest of the data in the lower tiers. The “Auto-Tier” policy automatically relocates data to the most appropriate tier based on the activity level of each data slice. Slices provisioned to a storage resource are relocated based on the highest performance disk drives available and the storage resource's slice temperature. Although this setting relocates data based on the storage resource's performance statistics, storage resources set with “Highest Available Tier” take precedence. When choosing Auto-Tier, the initial data placement is distributed across all available tiers. This distribution is based on the percentage of total available space that is in each tier. For example, if 70% of a pool's free capacity resides in the Capacity tier, 70% of the new slices are placed in that tier.

Start High then Auto-Tier (Recommended)

“Start High then Auto-Tier” is the default policy for each newly created storage resource. This is the recommended policy, because it takes advantage of “Highest Available Tier” for performance and “Auto-Tier” for TCO benefits. “Start High then Auto-Tier” sets the preferred tier for initial data allocation to the highest performing disk drives with available space, and then it relocates the storage resource’s data based on the performance statistics and the auto-tiering algorithm. This tiering policy maximizes initial performance and takes full advantage of the highest performing disks while improving TCO. With this tiering policy, less active data is moved to lower tiers, making room for more active data in the higher tiers. This tiering policy is especially good for storage resources that exhibit skew. As with the “Highest Available Tier” policy, setting this policy does not guarantee initial data placement in the highest tier. This only happens if there is available space not being used by other slices at the time of storage resource creation.

Lowest Available Tier

Use the “Lowest Available Tier” policy when cost effectiveness is the highest priority. With this policy, data is initially placed on the lowest available tier with capacity. Select this policy for storage resources that are not performance-sensitive or response-time-sensitive. Regardless of their activity level, all slices of these storage resources remain on the lowest available storage tier available in their pool. Slices with “colder” temperatures have higher priority to stay on the lowest tier if there is not enough space on this tier for all “Lowest Available Tier” data. When creating a storage resource with this policy, data is initially placed in the lowest available tier.

No Data Movement

If a storage resource is configured with this policy, no slices provisioned to the storage resource are relocated across tiers. Data remains in its current position, but can still be relocated within a tier. The system collects statistics on these slices after the tiering policy is changed. You can only apply the “No Data Movement” policy after a storage resource is created. Because of this, the initial data placement is set to the policy selected during creation.
Summary of tiering policies

Table 4. Tiering Policies below summarizes the different tiering policies.

<table>
<thead>
<tr>
<th>Tiering Policy</th>
<th>Corresponding Initial Tier Placement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highest Available Tier</td>
<td>Highest Available Tier</td>
<td>Sets the preferred tier for the initial data placement and subsequent data relocation (if applicable) to the highest-performing disk drives with available space.</td>
</tr>
<tr>
<td>Auto-Tier</td>
<td>Optimize for Pool Performance</td>
<td>Sets the initial data placement to Optimized for Pool and then relocates the LUN’s data based on the LUN’s performance statistics such that data is relocated among tiers according to I/O activity.</td>
</tr>
<tr>
<td>Start High then Auto-Tier</td>
<td>Highest Available Tier</td>
<td>First sets the preferred tier for the initial data placement to the highest-performing disk drives with available space, then relocates the LUN’s data based on the LUN’s performance statistics and the auto-tiering algorithm.</td>
</tr>
<tr>
<td>Lowest Available Tier</td>
<td>Lowest Available Tier</td>
<td>Sets the preferred tier for the initial data placement and subsequent data relocation (if applicable) to the most cost-effective disk drives with available space.</td>
</tr>
<tr>
<td>No Data Movement</td>
<td>Based on previous FAST VP policy used</td>
<td>Prevents any ongoing data movement for this LUN. The current placement of data is retained. Only available as a LUN property after the LUN is created.</td>
</tr>
</tbody>
</table>

Table 4. Tiering Policies

Managing FAST VP at the storage pool level

After you create storage resources with the most appropriate tiering policy, you can view and manage FAST VP at the storage pool level. When viewing the properties of a specific pool, the Unisphere FAST VP tab provides summary information about the amount of data targeted to be moved up, down, and within tiers. This tab also provides the ability to choose whether to include the pool in scheduled data relocations.
Figure 9. Storage pool tiering properties shows these properties.

In the storage pool details FAST VP tab, you can view both the relocation status and the tier details for the pool, and can choose whether to include the pool in scheduled relocations. Pools are included in scheduled relocations by default. If the box is unchecked, you must manually start the data relocations on the selected storage pool if you want FAST VP to run on the pool. Check the box to have the system perform data relocation based on the settings defined in the relocation schedule. Click the FAST VP Settings link to bring up the FAST VP Settings page and view this schedule. You can also access this page by navigating to Settings > Data Efficiency Settings and clicking on the Schedule tab.

The FAST VP tab displays the following information:

- Whether or not the pool is to be included in scheduled data relocations
- Data Relocation Status information:
  - **Paused** - Data relocation is paused for the system.
  - **Active** - Pool is actively relocating.
  - **Not Started** - Data relocation has not been started on the pool.
  - **Completed** - Data relocation has completed for the pool.
  - **Stopped by user** - Data relocation was stopped on this pool by the user.
  - **Failed** - Data relocation for the pool failed.
  - **Not Applicable** - No applicable data relocation status exists for the pool.
- Quantity of pool data, in GB, that is ready to move up, down, or be rebalanced within all storage tiers. This is the amount of data to move during the next scheduled data relocation. This number changes, since a recalculation is done right before the relocation window starts. If the Data Relocation Status is *Active*, this quantity is the amount of data to be moved during the current data relocation.

- Estimated time for the data relocation based on the data gathered by the software.

- Start and end times of the last data relocation.

- Data distribution per tier, including the quantity of free and total capacity.

- Amount of data that is ready to be moved up, down, or redistributed within the tier

**Relocation schedule**

The FAST VP feature allows for automatic data relocation based on a user-defined relocation schedule. This schedule defines when and how frequently the array starts data relocation on the participating storage pools. These scheduled relocations, which take place on all pools concurrently, can be arranged to take place during off-hours to minimize any adverse performance impact. The **Schedule** tab of the FAST VP Settings page allows you to change the data relocation schedule. You can launch this window from the storage pool properties **FAST VP** tab or by navigating to **Settings > Data Efficiency Settings** and clicking on the **Schedule** tab.
Figure 10. FAST VP Settings Schedule tab depicts the **Schedule** tab.

![FAST VP Settings Schedule tab](image)

In the Unisphere FAST VP **Schedule** tab, you can view whether scheduled relocation is enabled, and the current relocation schedule. You can select days of the week for relocations to run, as well as start and end times. Multiple relocation operations can occur during this timeframe, and the software assigns this schedule to all participating pools. This allows FAST VP to run continuously as a low priority background task if desired. FAST VP can also be scheduled for a smaller timeframe as a higher priority task. This allows scheduling of data relocation for off-peak hours in an attempt to minimize the performance impact from relocation. In the example shown in Figure 10, data relocations are allowed to run each day of the week for an eight hour window, starting at 22:00 (10:00 PM) and concluding at 6:00 AM.
The **General** tab shown in Figure 11. FAST VP Settings General tab provides additional configuration options as well as information regarding the next scheduled relocation.

![FAST VP Settings General tab](image)

**Figure 11. FAST VP Settings General tab**

The **General** tab enables you to define the operational rate for the data relocation operation. Valid values are **Low**, **Medium** (default), or **High**, where **Low** has the least impact on system performance, and **High** has the most impact. This tab also gives you the ability to enable or disable scheduled relocations. By default, scheduled relocations are enabled. The system-wide amount of data to relocate per tier is shown in this tab, as well as the estimated time for the next scheduled data relocation to complete.
Manual relocation

You can initiate manual data relocation for a pool at any time using Unisphere or the UEM CLI, whether or not the pool is included in the auto-tiering schedule. FAST VP analyzes all gathered statistics prior to beginning the relocation to ensure that up-to-date statistics and settings are properly accounted for prior to relocation. If the relocation completes before the end of the defined timeframe, FAST VP recollects statistics and continues relocation until the timeframe closes.

When you select **Start Data Relocation** in Unisphere, you can set the data relocation rate and end time. The data relocation rate can be set to **Low**, **Medium**, or **High**. The default relocation rate set on the **FAST VP Settings** page is **Medium**. (See Figure 12. **Start Data Relocation**.) You can also set the relocation end time to specify when the data relocation will stop – that is, when the relocation window closes.

![Start Data Relocation](image)

**Figure 12. Start Data Relocation**

Although the automatic scheduler is an array-wide setting, the manual data relocation operation allows for immediate relocation to take place at the pool level. Common situations for initiating a manual data relocation include:

- When reconfiguring a pool, if LUN properties, such as the tiering policy change and the new priority structure, must be implemented immediately
- As part of a script for a finer-grained relocation schedule

**FAST VP operations**

FAST VP operates by relocating the most active data directly to the highest available tier (either the Extreme Performance or Performance tier). To ensure sufficient space in the higher tiers, relocations attempt to reclaim 10% free space in the tiers to allow for the new slice allocation that occurs when new storage resources are created or when thin storage
resources consume additional capacity. To reclaim this 10% headroom, the least active slices within each tier move to lower tiers (Performance or Capacity).

FAST VP works at a granularity of 256 MB and relocates data by moving the entire slice to the appropriate tier, depending on the tiering policy selected for that particular storage resource.

**FAST VP algorithm**

FAST VP uses three different strategies to improve performance, capacity, and TCO. These techniques help identify and relocate slices to the most appropriate tiers by collecting statistics on each slice, analyzing the data, and relocating each slice based on its activity level.

**Statistics collection**

As previously noted, a slice of data is considered “hotter” (more active) or “colder” (less active) than another slice of data based on the relative activity level of those slices. The activity level of a particular slice is determined by counting the number of IOs (reads and writes) bound for each slice. FAST VP maintains a cumulative IO count and “weighs” each IO according to how recently it arrived. This weight deteriorates over time, and newer IOs are given a higher weight. After approximately 24 hours, the weights of IOs are nearly cut in half and continue to decrease. This statistics collection occurs continuously in the background for all storage resources.

**Analysis**

FAST VP analyzes the collected data once per hour. This analysis process produces a ranking order from “hottest” to “coldest” for each slice within the pool. Before relocation is invoked with automatic or manual relocation, FAST VP creates a candidate list of slices to move up, down, and within a pool. The ranking of a storage resource and its slices can be influenced by changing the tiering policy, in which case the tiering policy takes precedence over the activity level.

**Relocation**

During the user-defined relocation window, FAST VP promotes slices according to the candidate list that it created in the analysis stage. The hottest slices are moved to the highest tier available. During relocation, FAST VP prioritizes relocating slices to higher tiers. Slices are generally only relocated to lower tiers if the space they occupy is required for a higher priority slice. This way, FAST VP ensures that the higher performing disks are always used.

However there are exceptions to this rule. Slices set to “Lowest Available Tier” may be actively relocated to a lower tier if they are not there already. Also, as mentioned previously, FAST VP actively demotes the “coldest” slices in a tier if needed to free up 10% of capacity in a tier.
After data is added to a pool, FAST VP attempts to move it to higher tiers if space is available and the tiering policy calls for it. The relocation process aims to use all but 10% of the capacity in each tier. This creates space for any new slice allocations of higher priority storage resources before the next relocation. Lower tiers are used for capacity as the need arises. This entire relocation process is done automatically based on the user-defined relocation schedule, or manually, if user-initiated. Figure 13 provides an illustration of how FAST VP can improve slice placement for a pool.

As Figure 13. **FAST VP slice relocation** shows, before FAST VP is installed on the array, data ranging from very active to not active at all resides in a pool across each tier. The more expensive and higher performing disks may contain non-active data that would be more appropriate in the Capacity tier, for which the cost per GB is less expensive. Once FAST VP is installed the array automatically collects and analyzes statistics, and relocates data across tiers to better utilize the more expensive disks. FAST VP moves the most active data to a higher tier and the least active data to a lower tier.
Rebalance

Upon the expansion of a storage pool, the system recognizes the newly added space and initiates an auto-tiering data relocation operation called rebalancing. A rebalance relocates slices within the whole pool to achieve the best performance. If the pool is heterogeneous, relocation happens not only within each tier, but also across tiers. Pool expansions will initiate a rebalance regardless of whether the FAST VP license is installed.

When a rebalance is initiated and running, Unisphere displays a percentage number to indicate the progress as shown in Figure 14. *Rebalance in progress*. The example in Figure 14 shows a rebalance operation that is 12% complete.

![Rebalance in progress](image)

There is no relocation window for a rebalance operation, which runs until all slices in the relocation candidate list have been processed.

Load-balance

In addition to relocating slices across tiers based on relative slice temperature, FAST VP can also relocate slices within a tier to achieve maximum performance across disks in that tier. Some disks within a tier may be more heavily used than others. To improve performance, data slices are relocated within the tier to balance the load, if the tier is not 100% full. Maintaining 5% or more of free space allows for load-balancing to work efficiently.
Comparison and Usage of Multicore FAST Cache and FAST VP

The two features of the VNXe FAST Suite – FAST Cache and FAST VP – are designed to work together and complement each other to create a storage system that is optimized for flash technology.

FAST Cache allows the storage system to provide flash performance to the most heavily accessed chunks of data across the entire system. FAST Cache absorbs IO bursts from applications, thereby reducing the load on back-end hard disks. This improves the performance of the storage solution.

FAST VP, on the other hand, optimizes TCO by moving data to the appropriate storage tier based on sustained data access and demands over time.

Table 2. Multicore FAST Cache and FAST VP comparison compares the Multicore FAST Cache and FAST VP features.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Multicore FAST™ Cache</th>
<th>FAST™ VP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Allows FAST Cache SSDs to be used as a large capacity secondary cache.</td>
<td>Allows a single LUN to leverage the advantages of multiple disk types through the use of storage tiers.</td>
</tr>
<tr>
<td>Granularity</td>
<td>Granularity is 64 KB.</td>
<td>Granularity is 256 MB.</td>
</tr>
<tr>
<td>Operation</td>
<td>Data that is accessed frequently is copied from HDDs to the FAST Cache disks</td>
<td>Data is moved between different storage tiers based on weighted-average-of-access statistics collected over a period of time.</td>
</tr>
<tr>
<td>Timing</td>
<td>Constantly promotes frequently accessed HDD data to the FAST Cache. There are no relocation cycles.</td>
<td>Data movement occurs in scheduled or manually invoked relocation windows.</td>
</tr>
<tr>
<td>Analysis</td>
<td>Real-time monitoring decides which data needs to be promoted to the FAST Cache.</td>
<td>Hourly analysis decides which portion of data needs to be moved.</td>
</tr>
</tbody>
</table>

Table 2. Multicore FAST Cache and FAST VP comparison

When used together, the FAST Cache and FAST VP deliver high performance and efficiency and well as improved TCO for the storage system. For example, in scenarios where limited flash disks are available, the flash disks can be used to create the FAST Cache, and FAST VP can be used on a two-tier pool (Performance and Capacity).

From a performance point of view, the FAST Cache dynamically provides performance benefits to bursts of data, while FAST VP automatically moves “hotter” data to higher tiers. From a TCO perspective, FAST Cache, with a small number of flash disks, serves the data that is accessed most frequently, while FAST VP optimizes disk utilization and efficiency.
A general recommendation for the initial deployment of flash disks in a storage system is to use them for the FAST Cache first, since it benefits the entire system instead of only the single pool containing the flash disks. FAST Cache also dynamically assigns data to flash disks more quickly than FAST VP. This makes FAST Cache the most cost-efficient way to apply flash disks to a small system or a small active data set.

The FAST Cache feature is storage-tier-aware and works with FAST VP to make sure that storage system resources are not wasted by unnecessarily copying data to the FAST Cache, if that data is already on flash storage. If FAST VP moves a chunk of data to the Extreme Performance Tier, the system will not promote that chunk of data into FAST Cache, even if FAST Cache promotion criteria are met.

Similarly, if a chunk on a FAST-Cache-enabled storage resource receives a burst of IO, FAST VP does not immediately move that chunk to a higher storage tier. Instead, the system promotes the chunk into FAST Cache. The chunk’s activity level while in the FAST Cache will not be considered by FAST VP, which only looks at storage resource activity. However, the FAST Cache will eventually write this data back to the HDDs in order to make space for possible new promotions. Every time a page is copied to the storage resource (cleaned), FAST VP takes this IO activity taken into account. If data in the FAST Cache is copied to a storage resource frequently enough, due to a sustained application workload increase requiring new promotion space, FAST VP will eventually schedule a relocation of the storage resource’s data to a higher (possibly flash) tier.

**Interoperability**

The FAST Cache and FAST VP features are designed to work not only with each other, but also with all other VNXe features. This integration allows for ease of management, as users do not have to worry about conflicts when using multiple data services or features.

**Unified Snapshots**

The Unified Snapshots feature is fully compatible with the FAST Cache and FAST VP features. Using the Redirect on Write technology, changes to a snapped storage resource are written to a new location in the same pool. These new blocks will have the same Tiering Policy and initial allocation rules as the source storage resource because of the nature of block sharing between source storage resource and snapshots. This also applies to changes made to writable snapshots. These new blocks can be promoted and relocated the same as regular storage resource blocks. For more information, refer to the EMC VNXe3200 Unified Snapshots white paper on EMC Online Support.
**Multicore Cache**

As discussed previously, the FAST Cache and FAST VP algorithms only consider storage resource hits when identifying new data to promote or relocate. This means these algorithms do not count IOs to data in the Multicore Cache (DRAM Cache). This prevents data movement in situations where it may not be beneficial. However, a storage resource hit does occur when a page is copied to the storage resource during the DRAM Cache page cleaning process.

The FAST Cache and DRAM Cache are tightly integrated with one another. Both perform specific functions and supplement each other. The DRAM Cache shields the FAST Cache from high-frequency access patterns which are better served by the faster DRAM storage. With features such as prefetching, coalescing, and write throttling, the DRAM Cache is optimized for sequential IO. This allows the FAST Cache to concentrate on its own ideal workload, random IO. The two features have their own strengths, and they allow each other to service corresponding IO, improving overall system performance. Table 3 below summarizes the differences between the Multicore DRAM Cache and FAST Cache.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Multicore DRAM Cache</th>
<th>Multicore FAST Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Position</strong></td>
<td>Closest to the CPU, and has the lowest latency.</td>
<td>A step further from the CPU and is slower than DRAM cache.</td>
</tr>
<tr>
<td><strong>Best suited for</strong></td>
<td>Sequential IO</td>
<td>Random IO smaller than 64 KB High locality of reference</td>
</tr>
<tr>
<td></td>
<td>IO larger than 64 KB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Zero fill requests</td>
<td></td>
</tr>
<tr>
<td></td>
<td>High-frequency access patterns</td>
<td></td>
</tr>
<tr>
<td><strong>Response time</strong></td>
<td>Response time is from nanoseconds to microseconds.</td>
<td>Response time is from microseconds to milliseconds.</td>
</tr>
<tr>
<td><strong>Operation</strong></td>
<td>Has a single area that serves read and write operations.</td>
<td>Has a single area that serves read and write operations.</td>
</tr>
<tr>
<td><strong>Capacity</strong></td>
<td>Is limited in size (24 GB per Storage Processor) compared to the FAST Cache.</td>
<td>Can scale to larger capacities. Capacity is based on capacity of disks used.</td>
</tr>
<tr>
<td><strong>Granularity</strong></td>
<td>Each cache page is fixed at a size of 8 KB.</td>
<td>Each cache page is fixed at a size of 64 KB.</td>
</tr>
<tr>
<td><strong>Availability</strong></td>
<td>In case of failure, memory modules are customer-replaceable.</td>
<td>In case of failure, another flash disk hot spare automatically replaces the failing disk, and the faulted component is customer-replaceable.</td>
</tr>
<tr>
<td><strong>Power failure</strong></td>
<td>Contents are volatile and cannot withstand a power loss. Battery backup units sustain power while dirty pages are dumped to an internal mSATA disk.</td>
<td>Contents are non-volatile and can withstand a power loss.</td>
</tr>
</tbody>
</table>

*Table 3. Multicore DRAM Cache and Multicore FAST Cache comparison*
File-Level Retention (FLR)
The File-Level Retention feature is fully compatible with the FAST Cache and FAST VP features. File-Level Retention is a software feature that protects files from modification or deletion until a specified retention date. It is commonly used when regulations require maintaining data for a certain period of time. FAST VP can help to ensure that this infrequently accessed data does not take up space on higher tiers. This can either be done automatically by allowing FAST VP to auto-tier the data, or manually by setting FLR data to the “Lowest Available Tier” policy. For more information, refer to the *Using a VNXe System with CIFS File Systems* white paper on EMC Online Support.

File deduplication and compression
File deduplication and compression is fully compatible with the FAST Cache and FAST VP features. However, it is unlikely that deduplicated or compressed data will ever be in FAST Cache. This is because of policy criteria stating that a file must not be accessed for a certain period of time before it is eligible to be deduplicated and compressed. Due to this infrequent access, deduplicated and compressed data is likely to be relocated to lower tiers by FAST VP. For more information, refer to the *EMC VNXe File Deduplication and Compression* white paper on EMC Online Support.

Thin storage resources
Thin LUNs, file systems, and VMware datastores are fully compatible with the FAST Cache and FAST VP features. As new slices are allocated, they are placed according to the initial allocation rules of the appropriate tiering policy, as discussed in the Tiering Policies section of this paper. If the Tiering Policy was changed to “No Data Movement” after creation, new slices are placed according to the policy selected during creation. Using thin storage resources also allows you to maintain a concentrated set of active data in a pool, allowing FAST Cache and FAST VP to work more effectively.

Conclusion
The FAST Suite optimizes VNXe3200 storage systems for flash, which greatly benefits TCO and overall system performance. The VNXe FAST Suite makes the most efficient use of flash disk capacity by using flash disks for the most frequently accessed data in the storage system instead of dedicating disks to particular applications. The FAST Cache and FAST VP features achieve the goal of flash optimization in different ways, and are best suited to different situations.

The FAST Cache quickly handles dynamic and unpredictable workload changes, while FAST VP intelligently tiers data to the appropriate disks based on statistics collection and analysis over time. These two solutions are complementary, because they work on different granularity levels and with different frequencies. Implementing both FAST VP and FAST Cache can significantly improve performance and reduce storage costs.
The FAST Cache and FAST VP features optimize system performance automatically, requires minimal set up and can be easily managed through Unisphere. With the VNXe FAST Suite's intelligent data movement, storage administrators no longer have to worry about manually allocating the highest performance disks to the most IO intensive data. By facilitating the movement of active data to flash storage, FAST Cache and FAST VP allow users to take full advantage of the improved performance of flash technology.

References
The following white papers can be found on EMC Online Support:
- *EMC VNXe3200 Unified Snapshots*
- *EMC VNXe3200 File Deduplication and Compression*
- *Using a VNXe System with CIFS File Systems*