



White Paper

Optimizing Storage Performance and Cost with Intelligent Caching

Edited by Mark Woods

August 2010 | WP-7107

ABSTRACT

NetApp is a pioneer in the development of innovative intelligent caching technologies to decouple storage performance from the number of disks in the underlying disk array to substantially improve cost. This white paper describes the NetApp® implementation of write caching using NVRAM as a journal, as well as a three-level approach to read caching that includes the system buffer cache, Flash Cache, and FlexCache®. These technologies yield significant performance and cost improvements for a wide range of real-world environments and applications.

TABLE OF CONTENTS

1	EXECUTIVE SUMMARY	3
2	INTRODUCTION	3
3	INTELLIGENT CACHING OF WRITE REQUESTS	3
	JOURNALING WRITE REQUESTS	4
	OPTIMIZING WRITE PERFORMANCE	4
4	NETAPP READ CACHING TECHNIQUES	5
	READ CACHING IN SYSTEM MEMORY	6
	FLASH CACHE	7
	FLEXCACHE	8
5	CACHING AND PRIORITY OF SERVICE	9
6	CACHING AND STORAGE EFFICIENCY	10
7	REAL-WORLD APPLICATIONS OF NETAPP INTELLIGENT CACHING	11
	SERVER AND DESKTOP VIRTUALIZATION	12
	CLOUD COMPUTING	12
	REMOTE OFFICE	12
	DATABASE	12
	E-MAIL	13
	FILE SERVICES	13
	ENGINEERING AND TECHNICAL APPLICATIONS	13
8	CONCLUSION	14
9	REFERENCES	15
10	ACKNOWLEDGMENTS	15

1 EXECUTIVE SUMMARY

The intelligent use of caching technologies provides a way to decouple storage performance from the number of disks in the underlying disk array to substantially improve cost. NetApp has been a pioneer in the development of innovative read and write caching technologies. NetApp storage systems use NVRAM to journal incoming write requests, allowing NetApp to commit write requests to nonvolatile memory and respond back to writing hosts without delay. Caching writes early in the stack (most other storage vendors cache writes at the device driver level) allows NetApp to optimize writes to disk, even when writing to double-parity RAID. NetApp sees only a 2% drop in performance relative to its single-parity RAID, while at least one competing vendor sees a 30% write performance drop.

NetApp uses a multilevel approach to read caching. The first-level read cache is provided by the system buffer cache. Special algorithms decide which data to retain in memory and which data to prefetch to optimize this function. NetApp Flash Cache provides an optional second-level cache, accepting blocks as they are ejected from the buffer cache to create a large, low-latency block pool to satisfy read requests. Flash Cache can reduce your storage costs by 50% or more by reducing the number of spindles needed for a given level of performance and allowing you to replace high-performance disks with more economical options.

Both buffer cache and Flash Cache benefit from a cache amplification effect that occurs when NetApp deduplication or FlexClone® technologies are used. Behavior can be further tuned and priorities can be set using NetApp FlexShare® to create different classes of service.

The third-level read cache is provided by NetApp FlexCache, which creates a separate caching tier in your storage infrastructure, scaling read performance beyond the boundaries of a single storage system's capabilities.

These technologies demonstrate significant improvements in a wide range of real-world environments and applications, including infrastructure virtualization, cloud computing, remote office, database, e-mail, file services and a variety of engineering and technical applications such as EDA, PLM, rendering, oil and gas exploration, and software development.

2 INTRODUCTION

Traditionally, storage performance has been closely tied to spindle count; the primary means of boosting storage performance was to add more or higher performance disks. However, the intelligent use of caching can dramatically improve storage performance for a wide variety of applications.

From the beginning, NetApp has pioneered innovative approaches to both read and write caching that allow you to do more with less hardware and at less cost. This white paper explains how NetApp caching technologies can help you:

- Increase I/O throughput while decreasing I/O latency (the time needed to satisfy an I/O request)
- Decrease storage capital and operating costs for a given level of performance
- Eliminate much of the manual performance tuning that is necessary in traditional storage environments

The use of NetApp intelligent caching technologies with a variety of real-world applications is also discussed.

3 INTELLIGENT CACHING OF WRITE REQUESTS

Caching writes has been used as a means of accelerating write performance since the earliest days of storage. NetApp uses a highly optimized approach to write caching that integrates closely with the NetApp Data ONTAP® operating environment to eliminate the need for the huge and expensive write caches seen on some storage arrays, while enabling NetApp to achieve exceptional write performance, even with RAID 6—double-parity RAID.

JOURNALING WRITE REQUESTS

When any storage system receives a write request, it must commit the data to permanent storage before the request can be confirmed to the writer. Otherwise, if the storage system were to experience a failure with data only in volatile memory, that data would be lost, and underlying file structures could become corrupted.

Storage system vendors commonly use battery-backed, nonvolatile RAM (NVRAM) to cache writes and accelerate write performance while providing permanence, because writing to memory is much faster than writing to disk. NetApp provides NVRAM in all of its current storage systems, but the NetApp Data ONTAP operating environment uses NVRAM in a much different manner than typical storage arrays.

Every few seconds, Data ONTAP creates a special Snapshot™ copy called a consistency point, which is a completely consistent image of the on-disk file system. A consistency point remains unchanged even as new blocks are being written to disk because Data ONTAP does not overwrite existing disk blocks. The NVRAM is used as a journal of the write requests that Data ONTAP has received since creation of the last consistency point. With this approach, if a failure occurs, then Data ONTAP simply reverts to the latest consistency point and replays the journal of write requests from NVRAM in order to bring the system up to date and make sure the data and metadata on disk are current.

This is a much different use of NVRAM than that of traditional storage arrays, which cache write requests at the disk driver layer, and it offers several advantages.

- **Requires less NVRAM.** Processing a write request and caching the resulting disk writes generally take much more space in NVRAM than simply journaling the information required to replay the request. Consider a simple 8KB NFS write request. Caching the disk blocks that have to be written to satisfy the request requires 8KB for the data, 8KB for the inode, and—for large files—another 8KB for the indirect block. Data ONTAP merely has to log the 8KB of data along with about 120 bytes of header information, so it uses half or a third as much space for the same operation.

It's common for other vendors to point out that NetApp storage systems often have far less NVRAM than competing models. This is because NetApp storage systems actually need less NVRAM to do the same job because of their unique use of NVRAM.

- **Decreases the criticality of NVRAM.** When NVRAM is used as a cache of unwritten disk blocks, it becomes an integral part of the disk subsystem. A failure can cause significant data corruption. If something goes wrong with the NVRAM in a NetApp storage system, a few write requests might be lost, but the on-disk image of the file system will remain completely self-consistent.
- **Improves response times.** Both block-oriented SAN protocols (Fibre Channel protocol, iSCSI, FCoE) and file-oriented NAS storage protocols (CIFS, NFS) require an acknowledgement from the storage system that a write has been completed. To reply to a write request, a storage system without any NVRAM must update its in-memory data structures, allocate disk space for new data, and wait for all modified data to reach disk. A storage system with an NVRAM write cache performs the same steps but copies modified data into NVRAM instead of waiting for disk writes. Data ONTAP can reply to a write request much more quickly because it need only update its in-memory data structures and log the request. It does not have to allocate disk space for new data or copy modified data and metadata to NVRAM.
- **Optimizes disk writes.** Journaling all write data immediately and acknowledging the client or host not only improve response times, but also give Data ONTAP more time to schedule and optimize disk writes. Storage systems that cache writes in the disk driver layer necessarily have to accelerate processing in all the intervening layers in order to provide a quick response to host or client, giving them less time to optimize. (This is described further in the following section.)

You can learn more about how Data ONTAP benefits from NVRAM and NetApp WAFL® (Write Anywhere File Layout) in [NetApp technical report 3001](#). This is the original technical report that was written when NetApp was founded. (It was refreshed in 2006.)

OPTIMIZING WRITE PERFORMANCE

No matter how big a write cache is or how it is used, eventually data has to be written to disk. Data ONTAP divides its NVRAM into two separate buffers. When one buffer is full, that triggers disk write activity to flush

all the cached writes to disk and create a consistency point. Meanwhile, the second buffer continues to collect incoming writes until it is full, and then the process reverts to the first buffer. This approach to caching writes—in combination with WAFL—is closely integrated with NetApp RAID 4 and RAID-DP® and allows NetApp to schedule writes such that disk write performance is optimized for the underlying RAID array. The combination of NetApp NVRAM and WAFL in effect turns a set of random writes into sequential writes.

In order to write new data into a RAID stripe that already contains data (and parity), you have to read the parity block and calculate a new parity value for the stripe and then write the data block plus the new parity block. That's a significant amount of extra work required for each block to be written.

NetApp reduces this penalty by buffering NVRAM-protected writes in memory and then writing full RAID stripes plus parity whenever possible. This makes reading parity data before writing unnecessary and only requires a single parity calculation for a full stripe of data blocks. WAFL does not overwrite existing blocks when they are modified, and it can write data and metadata to any location. In other data layouts, modified data blocks are usually overwritten, and metadata is often usually required to be at fixed locations.

This approach offers much better write performance, even for double-parity RAID (RAID 6). Unlike other RAID 6 implementations, NetApp RAID-DP performs so well that it is the default option for NetApp storage systems. Tests show that random write performance declines only 2% versus the NetApp RAID 4 implementation. By comparison, another major storage vendor's RAID 6 random write performance decreases by 33% relative to RAID 5 on the same system. (RAID 4 and RAID 5 are both single-parity RAID implementations. RAID 4 uses a designated parity disk. RAID 5 distributes parity information across all disks in a RAID group.)

Full technical details of NetApp RAID-DP are described in [NetApp technical report 3298](#). You can read more about the advantages of RAID-DP versus other RAID options in [NetApp white paper 7005](#).

4 NETAPP READ CACHING TECHNIQUES

The random read performance of a storage system is dependent on both drive count (total number of drives in the storage system) and drive rotational speed. Unfortunately, adding more drives to boost storage performance also means using more power, more cooling, and more space, and—with single disk capacity growing much more quickly than performance—many applications require additional disk spindles to achieve optimum performance even when the additional capacity is not needed.

Read caching is the process of deciding which data to either keep or prefetch into storage system memory in order to satisfy read requests more rapidly. NetApp uses a multilevel approach to read caching to break the link between random read performance and spindle count, providing you with multiple options to deliver low read latency and high read throughput while minimizing the number of disk spindles you will require:

- **Read caching in system memory** (the system buffer cache) provides the first-level read cache and is utilized in all current NetApp storage systems.
- **Flash Cache** (PAM II) provides an optional second-level read cache to supplement system memory.
- **FlexCache** creates a separate caching tier within your storage infrastructure to satisfy read throughput requirements in the most data-intensive environments.

The system buffer cache and Flash Cache increase read performance within a storage system. FlexCache scales read performance beyond the boundaries of any single system's performance capabilities.

NetApp deduplication and other storage efficiency technologies eliminate duplicate blocks from disk storage and thus make sure that valuable cache space is not wasted storing multiple copies of the same data blocks. Both the system buffer cache and Flash Cache benefit from this “cache amplification” effect. The percentage of cache hits increases and average latency improves as more shared blocks are cached. NetApp FlexShare software can also be used to prioritize some workloads over others and modify caching behavior to meet specific objectives. This is discussed further in section 4.

READ CACHING IN SYSTEM MEMORY

There are two distinct aspects to read caching:

- Keeping “valuable” data in system memory
- Prefetching data into system memory before it is requested

DECIDING WHICH DATA TO KEEP IN SYSTEM MEMORY

The simplest means of accelerating read performance is to cache data in system memory once it arrives there. If another request for the same data is received, that request can then be satisfied from memory rather than having to reread it from disk. However, for each block in the system buffer cache, Data ONTAP must determine the potential “value” of the block. The questions that must be addressed for each data block include:

- Is the data likely to be reused?
- How long will/should the data stay in memory?
- Will the data change before it can be reused?

Answers to these questions can be determined in large part based on the type of data and how it got into memory in the first place.

- **Write data.** Write workloads tend not to be read back after writing (they are often already cached locally on the system that performed the write), so they’re generally not good candidates for caching. In addition, recently written data is normally *not* a high priority for retention in the system buffer cache because the overall write workload can be high enough that writes overflow the cache and cause other more valuable data to be ejected. However, some read-modify-write type workloads benefit from caching recent writes. Examples include stock market simulations and some engineering applications.
- **Sequential reads.** Sequential reads can often be satisfied by reading a large amount of contiguous data from disk at one time. In addition, as with writes, caching large sequential reads can cause more valuable data to be ejected from system cache. Therefore, it’s actually preferable to read such data from disk and preserve available read cache for data that is more likely to be read again. NetApp also provides algorithms to recognize sequential read activity and read data ahead (read-ahead algorithms will be discussed shortly), making it unnecessary to retain this type of data in cache with a high priority.
- **Metadata.** Metadata describes where and how data is stored on disk (name, size, block locations, and so on). Because metadata is needed to access user data, it is normally cached with high priority in order to avoid the need to read metadata from disk before every read and write.
- **Small, random reads.** Small, random reads are the most expensive disk operation because they require a higher number of head seeks per kilobyte than sequential reads. Head seeks are a major source of the read latency associated with reading from disk. Therefore, data that is randomly read is a high priority for caching in system memory.

The default caching behavior for the Data ONTAP buffer cache is to prioritize small, random reads and metadata over writes and sequential reads.

DECIDING WHICH DATA TO PREFETCH INTO SYSTEM MEMORY

The NetApp read ahead algorithms are designed to anticipate what data will be requested next and read it into memory before the read request actually arrives. Because of the importance of effective read ahead algorithms, NetApp has done a significant amount of research in this area. Data ONTAP uses an adaptive read history logging system based on “[read sets](#),” which provides much better performance than traditional and fixed read-ahead schemes.

In fact, multiple read sets can support caching for individual files or LUNs, which means that multiple read streams can be prefetched simultaneously. The number of read sets per file or LUN object is related to the frequency of access and the size of the object.

The system adaptively selects an optimized read-ahead size for each read stream based on historical factors, such as:

- The number of read requests processed in the read stream
- The amount of host-requested data in the read stream
- A read access style associated with the read stream
- Forward and backward reading
- Identifying coalesced and fuzzy sequences of arbitrary read access patterns

Cache management is significantly improved by these algorithms, which determine when to perform read-ahead operations and how long each read stream's data should be retained in cache.

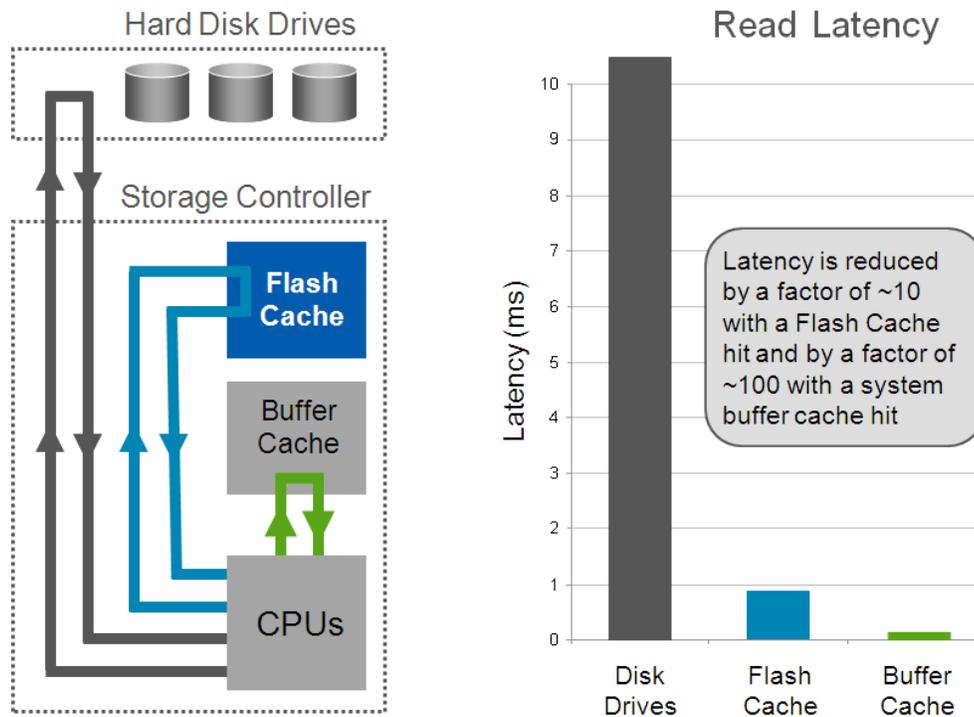


Figure 1) Impact of the system buffer cache and Flash Cache on read latency.

FLASH CACHE

The next level of read cache in NetApp storage systems is the optional Flash Cache (PAM II), which is designed to extend the capabilities of the system buffer cache described above and further optimize the performance of random read-intensive workloads such as file services, messaging, virtual servers and desktops (VDI), and OLTP databases without the need for added disk spindles. Flash Cache speeds access to your data, reducing latency by a factor of 10 or more compared to disk reads.

FLASH CACHE OPERATION

Flash Cache is a second-level cache: a cache used to hold blocks evicted from the system buffer cache. This allows the Flash Cache software to work seamlessly with the first-level read cache in system memory and its read-ahead mechanisms. As data flows from system memory, the priorities and categorization already performed on the data allow the Flash Cache to make decisions about what is or isn't accepted into the cache.

With Flash Cache, a storage system first checks to see whether a requested read has been cached in one of its installed modules before issuing a disk read. Data ONTAP maintains a set of cache tags in system

memory and can determine whether Flash Cache contains the desired block without accessing the cards, accelerating access to the Flash Cache and reducing latency. As with the system buffer cache, the key to success lies in the algorithms used to decide what goes into the cache.

By default, Flash Cache algorithms try to distinguish high-value, randomly read data from sequential and/or low-value data and maintain that data in cache to avoid time-consuming disk reads. NetApp also provides the ability to change the behavior of the cache to meet unique requirements. The three modes of operation are:

- **Default mode.** The normal mode of Flash Cache operation caches both user data and metadata, similar to the caching policy for the system buffer cache. For file service protocols such as NFS and CIFS, metadata includes the data required to maintain the file and directory structure. With SAN, the metadata includes the small number of blocks that are used for the bookkeeping of the data in a LUN. This mode is best used when the size of the active data set is equal to or less than the size of the Flash Cache. It also helps when there are hot spots of frequently accessed data and makes sure that the data will reside in cache.
- **Metadata mode.** In this mode only storage system metadata is cached. In some situations, metadata is reused more frequently than specific cached data blocks. Caching metadata can have a significant performance benefit and is particularly useful when the data set is very large or composed of many small files, or the active portion of the data set is very dynamic. Caching metadata might work well for data sets that are too large to be effectively cached (that is, the active data set exceeds the size of the installed cache). Metadata mode is the most restrictive mode in terms of what data is allowed in the cache.
- **Low-priority mode.** In low-priority mode, caching is enabled not only for “normal” user data and metadata but also for low-priority data that would normally be excluded. Low-priority data in this category includes large sequential reads and data that has recently been written. The large amount of additional cache memory provided by Flash Cache might allow sequential reads to be stored without negatively affecting other cached data. This is the least-restrictive operating mode for Flash Cache.

PREDICTIVE CACHE STATISTICS

NetApp has developed Predictive Cache Statistics (PCS) software to determine whether a storage system can benefit from added cache without requiring you to purchase or install the hardware. PCS allows you to model the impact of adding various amounts of Flash Cache to a storage system.

Using PCS, you can determine whether Flash Cache will improve performance for your workloads and decide how much additional cache you will need. PCS also allows you to test the different modes of operation to determine whether the default, metadata, or low-priority mode is best.

Full details of NetApp Flash Cache including PCS are provided in [TR-3832: Flash Cache and PAM Best Practices Guide](#).

FLEXCACHE

FlexCache, the third-level NetApp read cache, enables you to create a caching architecture within your storage infrastructure. This approach has already demonstrated significant benefits for accelerating the speed of parallel software builds as well as compute-intensive applications such as animation rendering, electronic design automation, seismic analysis, and financial market simulation. FlexCache can also improve read performance across wide area networks (WANs) and enables the migration of virtual machines over distance while maintaining service-level agreements (SLAs).

In fact, FlexCache can serve in almost any situation that would otherwise require a fan-out design—one where data must be replicated to multiple storage systems to boost performance—while offering:

- Data consistency and coherency.
- An adaptive cache that immediately responds to changing working sets.
- The ability to fully leverage the capabilities of the NetApp buffer cache and NetApp Flash Cache (as described above) as well as solid-state disk drives (SSDs) within a FlexCache node.

- A performance tier that allows your back-end storage to use less expensive, capacity-oriented SATA disk drives.
- Inherent thin provisioning. Fan out storage volumes require at least as much storage as the source volume. Because FlexCache only caches hot data, it requires only enough storage to accommodate the active working set.

FlexCache currently works with NFS Version 2 and NFS Version 3 and is provided as part of Data ONTAP at no additional cost and can be implemented in any existing NetApp storage system.

THE FLEXCACHE ARCHITECTURE

With the FlexCache architecture, a small, fast storage cache resides logically between your primary storage systems and your compute servers or clients. As with an internal read cache, data is automatically copied into this caching tier the first time it is read; subsequent reads are satisfied from the cache rather than the source storage system. By concentrating your investments in Flash Cache (PAM II) and SSDs in a centralized caching tier, you can leverage that investment across multiple applications and multiple storage systems behind the FlexCache tier. You can also deploy more economical, high capacity disk drives in your primary storage, improving your overall storage efficiency without affecting performance.

FlexCache eliminates storage bottlenecks without requiring additional administrative overhead for data placement. The addition of SSDs and/or read cache to a NetApp caching architecture will extend the benefits of FlexCache by reducing response times for critical applications, boosting transaction rates and cutting energy consumption in your caching layer. Like any storage system, the I/O performance of a caching appliance depends on the number of disk spindles. Flash-memory technologies—either SSDs or cache—will significantly reduce the number of disks needed to achieve a given performance level while also reducing response time.

You can learn more about FlexCache in [TR-3669: FlexCache Caching Architecture](#)

5 CACHING AND PRIORITY OF SERVICE

Data ONTAP includes priority of service capabilities that allow you to control how cache and other resources are allocated to different workloads as a standard feature. With NetApp FlexShare software you can dynamically prioritize storage traffic at the volume or LUN level to provide different classes of service. This means you can give higher priority to the volumes used by more critical workloads and safely support multiple applications of varying priorities on one storage system. There is no additional cost to take advantage of this feature.

FlexShare provides three parameters that can be configured independently on each volume to tune your storage to the needs of every application:

- **Relative priority.** A higher priority gives a volume a greater percentage of available resources (CPU, cache memory, I/O) when a system is fully loaded. If higher priority applications aren't busy, lower priority applications can use available resources without limitation.
- **User versus system priority.** Prioritize user workloads (application and end-user traffic) over system work (backup, replication, and so on) or vice versa.
- **Cache utilization.** Configure the cache to retain data in cache or reuse the cache depending on workload characteristics. Optimizing cache usage can significantly increase performance for data that is frequently read and/or written.

These parameters can be configured dynamically, so you can change all or some of the settings on the fly as your needs change, even if they change daily or hourly. Once FlexShare is configured, it directs the way storage system resources are used to provide an appropriate level of service to each application.

FlexShare is fully compatible not only with the NetApp buffer cache but also with Flash Cache. Settings made in FlexShare apply to the data in both caches. FlexShare allows even finer grained control to be applied on top of global policies. For example, if an individual volume is given a higher priority with FlexShare, data from that volume will receive a higher cache priority. You can also set different caching policies for each volume if desired. You can similarly implement FlexShare on systems utilizing FlexCache for finer grained control of caching within your caching architecture.

6 CACHING AND STORAGE EFFICIENCY

NetApp intelligent caching can improve your storage efficiency in two important ways:

- Certain NetApp storage efficiency features create a “cache amplification” effect for shared storage blocks that increases the benefits you derive from investments in cache.
- Intelligent caching allows you to use fewer and/or less expensive disks

Cache Amplification in a Virtual Infrastructure

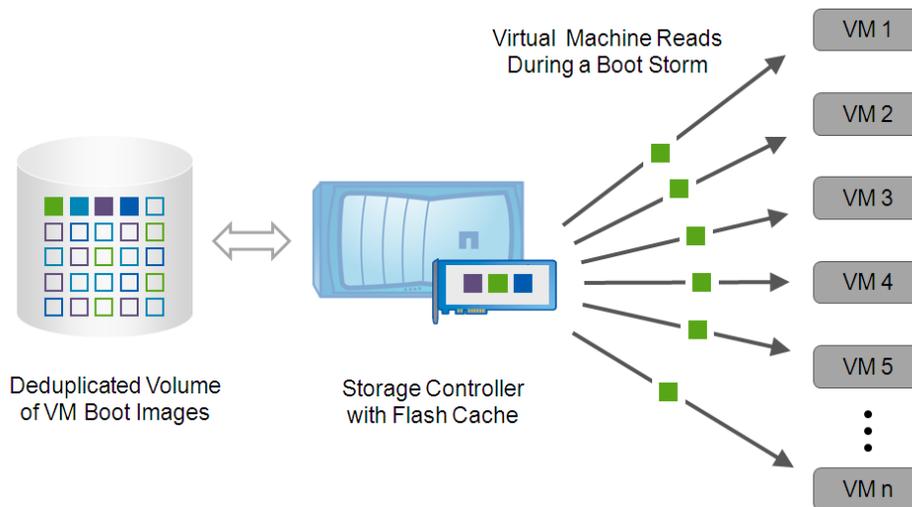


Figure 2) Cache amplification in a virtual infrastructure environment showing the advantage of having deduplicated blocks in cache. (This effect is also referred to as transparent storage cache sharing [TSCS].)

Many applications have high levels of block duplication. The result is that you not only end up wasting storage space storing identical blocks, you also waste cache space by caching these identical blocks in system buffer cache and Flash Cache. NetApp deduplication and NetApp FlexClone technology can enhance the value of caching by eliminating block duplication and increasing the likelihood that a cache hit will occur. Deduplication identifies and replaces duplicate blocks in your primary storage with pointers to a single block. FlexClone allows you to avoid the duplication that typically results from copying volumes, LUNs, or individual files—for example, for development and test operations. In both cases, the end result is that a single block could have many pointers to it. When such a block is in cache, the probability that it will be requested again is therefore much higher.

This cache amplification effect becomes particularly advantageous in conjunction with server and desktop virtualization, as we'll see in the following section. In that context, cache amplification has also been referred to as [transparent storage cache sharing](#) (TSCS) as an analogy to the transparent page sharing (TPS) of VMware® servers.

The use of Flash Cache and FlexCache can significantly decrease the cost of your disk purchases and make your storage environment more efficient. For example, testing in a Windows® file services environment showed:

- Combining Flash Cache with Fibre Channel or SAS disks can improve performance while using 75% fewer spindles and decreasing purchase price by 54%, while at the same time saving 67% on both power and space.

- Combining Flash Cache with SATA disks can deliver the same performance as Fibre Channel or SAS disks and more capacity while lowering cost per terabyte of storage by 57% while saving 66% on power and 59% on space.

Optimize Storage Efficiency and Performance with Flash Cache

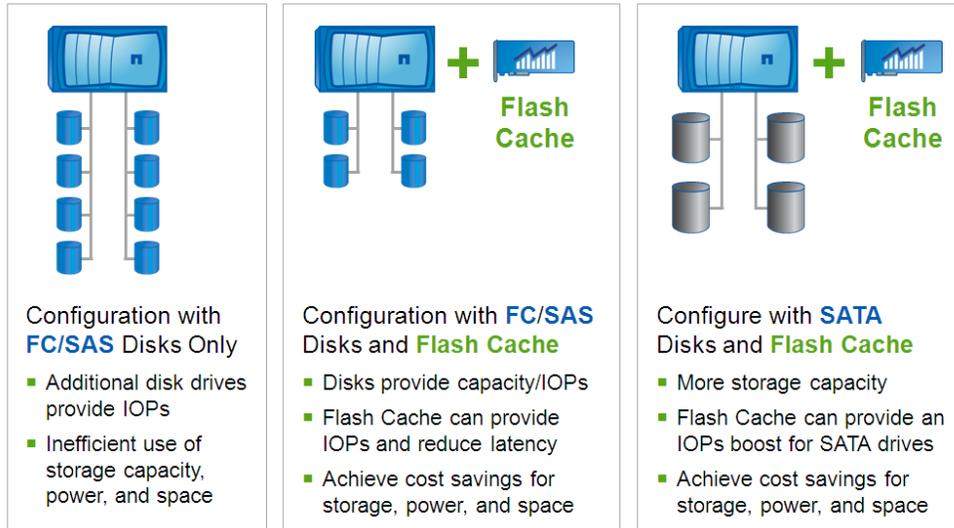


Figure 3) Impact of Flash Cache on storage efficiency.

7 REAL-WORLD APPLICATIONS OF NETAPP INTELLIGENT CACHING

A wide range of IT environments and applications can benefit from intelligent write and read caching. The default settings work extremely well in most cases, while certain applications benefit from further tuning of cache behavior or adjusting resource prioritization using NetApp FlexShare.

Table 1) Applicability of intelligent caching to various environments and applications. (Hyperlinks are to related references for each environment/application.)

Environment/Application	Write Caching	Read Cache	Flash Cache	FlexCache
Server/desktop virtualization	X	X	X	X
Cloud computing	X	X	X	X
Remote office	X	X		X
Database	X	X	X	
E-mail	X	X	X	
File services	X	X	X	X
Engineering and Technical Applications				
Product lifecycle management	X	X	X	
Oil and gas exploration	X	X	X	
Software development	X	X	X	X
Electronic design automation	X	X	X	X
Rendering	X	X	X	X

SERVER AND DESKTOP VIRTUALIZATION

Both server virtualization and virtual desktop infrastructure (VDI) create some unique storage performance requirements that caching can help to address. Any time you have a need to boot a large number of virtual machines at one time—for instance, during daily desktop startup or, in the case of server virtualization, after a failure or restart—you can create a significant storage load. Large numbers of logins and virus scanning can also create heavy I/O load.

For example, a regional bank had over 1,000 VMware View desktops and was seeing significant storage performance problems with its previous environment despite having 300 disk spindles. When that environment was replaced with a NetApp solution using just 56 disks plus Flash Cache (PAM II), outages due to reboot operations dropped from 4 to 5 hours to just 10 minutes. Problems with nonresponsive VDI servers simply went away, and logins, which previously had to be staggered, can now be completed in just 4 seconds. The addition of NetApp intelligent caching gave the bank more performance at lower cost.

As you move toward full virtualization across multiple data centers, the ability to transparently migrate virtual machines and their associated applications between data centers separated by substantial physical distances becomes desirable. Combining active storage at one site with [cached storage using NetApp FlexCache at a second site can facilitate long distance VM migration](#) for load balancing or other purposes. This approach might be particularly well suited to situations where the migration is only temporary since it can make it unnecessary to migrate the full data set.

Because of the high degree of duplication in virtual environments (that results from having many copies of the same operating systems and applications), they can experience an extremely high rate of cache amplification with shared data blocks. You can either eliminate duplication by applying NetApp deduplication to your existing virtual environment or—if you are setting up a new virtual environment—by using the NetApp Virtual Storage Console v2.0¹ provisioning and cloning capability to efficiently clone each virtual machine such that each similar virtual machine shares the same blocks. Either way, once the set of shared blocks has been read into cache, read access is accelerated for all virtual machines.

CLOUD COMPUTING

Since most cloud infrastructure is built on top of server virtualization, cloud environments will experience many of the same benefits from intelligent caching. In addition, the combination of intelligent caching and FlexShare lets you fully define classes of service for different tenants of shared storage in a multi-tenant cloud environment. This can significantly expand your ability to deliver IT as a service (ITaaS).

For example, you can put Flash Cache into metadata mode such that only metadata is cached. To cache user data for individual volumes, you assign them a FlexShare buffer cache setting of “keep.” Alternatively, you could leave Flash Cache in the default mode—metadata and user data are cached—and lower the priority for some volumes by setting the buffer cache policy to “reuse,” so that data from those volumes either will not be cached or will be evicted from cache first.

REMOTE OFFICE

In a distributed enterprise, it can be difficult to know what data will be needed in what location. Remote offices often have a need to access data from corporate data centers, but WANs can make such data access slow, reducing productivity.

These problems can be addressed using FlexCache in the remote location. Once a file is read across the WAN into a local FlexCache volume, it can be accessed repeatedly at local LAN speed. This works well for distributed organizations involved in application development and test where read performance is important. It also works well in organizations where users might need to change locations frequently. Such users can get accelerated access to home directories and other files in any location where FlexCache is in place.

DATABASE

Intelligent caching provides significant benefits in online transaction processing environments as well. [A recent NetApp white paper](#) examined two methods of improving performance in an I/O-bound OLTP

¹Virtual Storage Console v2.0 is a plugin for VMware vCenter™ Server Virtualization Management environment, which was jointly developed by NetApp and VMware.

environment: adding additional disks or adding Flash Cache (PAM II). Both approaches were effective at boosting overall system throughput. The Flash Cache configuration:

- Costs about 30% less than the same system with additional disk
- Reduces average I/O latency from 27.5 milliseconds to 16.9 milliseconds
- Consumes no additional power or rack space (the configuration with additional disk increases both by more than a factor of 2)

Database development environments can experience significant advantages by combining Flash Cache, high-capacity SATA disks, and NetApp FlexClone. [With FlexClone, you can quickly create space-efficient copies of your production databases for testing or other purposes.](#) Having multiple clones of the same database will result in significant cache amplification effect, enhancing performance.

E-MAIL

E-mail environments with large numbers of users quickly become extremely data intensive. As with database environments, the addition of Flash Cache can significantly boost performance at a fraction of the cost of adding more disks. For example, in recent NetApp benchmarking with Microsoft® Exchange 2010, the addition of Flash Cache doubled the number of IOPs achieved, increased the supported number of mailboxes by 67%, and improved storage efficiency as well. These results will be described in TR-3865: Using Flash Cache for Exchange 2010, scheduled for publication in September, 2010.

FILE SERVICES

File services environments can often benefit from the use of Flash Cache (PAM II) and/or FlexCache. For example, a premier Internet site for photos, videos, and graphics has 7 billion images and 46 million unique visitors per month. Storage had quickly become one of this company's largest expenses and biggest headaches, and data center space was at a premium.

A new storage architecture was designed to use fewer, more powerful storage controllers with PAM I caching modules (the first generation NetApp performance accelerator that preceded Flash Cache) and SATA disks instead of Fibre Channel disks. The final configuration offered an 8-to-1 reduction in total numbers of storage systems, offered the same performance, and reduced the footprint in the data center from 16 racks to just 4. The key to performance was to cache the metadata of active files in the added intelligent cache.

(See the following section for a description of a FlexCache configuration for high-performance read caching in rendering and other technical environments, which are in essence high-performance file serving environments.)

ENGINEERING AND TECHNICAL APPLICATIONS

A wide variety of applications, such as those for rendering, electronic design automation (EDA), seismic data analysis, and financial simulation, use large compute farms to process data. Multiple compute nodes must read the same data set, resulting in a storage bottleneck that typically can only be relieved by copying data to multiple storage systems. Using FlexCache, you can achieve the same result much more efficiently and automatically without risky duplication.

By distributing file access loads across multiple storage systems, FlexCache scales the overall performance of the compute farm without adding management complexity. Hot data sets are automatically replicated from the origin storage volume to FlexCache volumes on one or more caching storage systems. Additional caching systems can be added as needed to scale performance.

FlexCache is optimally efficient because it only transfers the data that is actually needed to the cache. As a result, the caching system uses storage space very efficiently. Less storage is required, providing significant cost savings versus alternatives that use data copies to relieve storage bottlenecks. As we've already seen, you can concentrate your investments in Flash Cache, fast disks, or SSDs in the caching tier to maximize your return on that investment while using lower cost, high-capacity disk drives in back-end storage systems to reduce overall storage costs.

For example, the visual effects industry uses large numbers of servers that must access the same storage in parallel to produce the stunning visual effects in today's motion pictures. The I/O volume generated in these "render farms" is often enough to overwhelm individual file servers. In practice, visual effects companies usually find it necessary to make multiple copies of hot data sets.

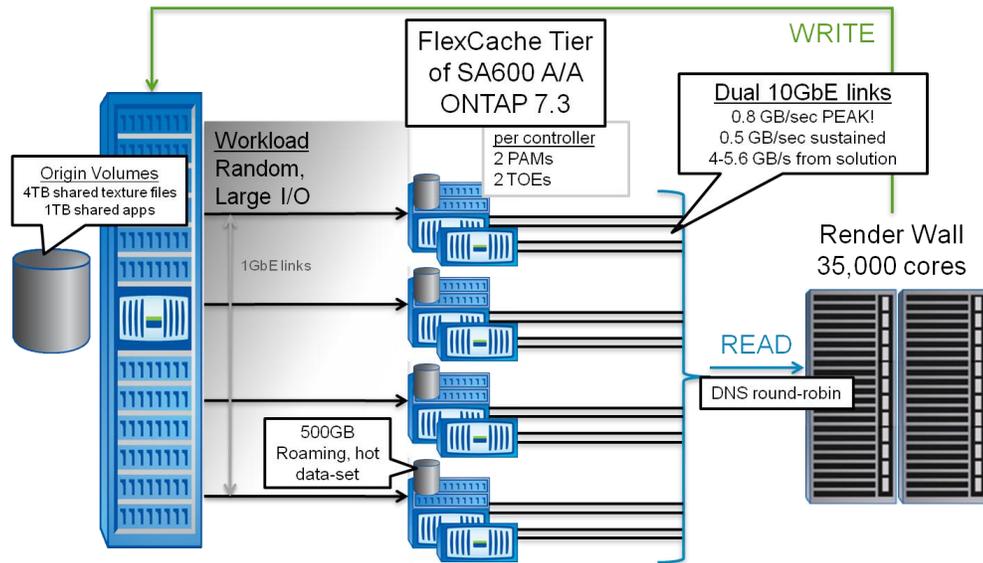


Figure 4) FlexCache deployment in a rendering environment.

A leading visual effects company solved this problem by replacing its manual, fan-out storage with a set of four NetApp SA600 caching appliances. This solution not only improved performance, but also adapted automatically to changes in usage—always caching the active data set—and eliminated the headaches associated with manually managing replication, as described in [a recent article](#).

8 CONCLUSION

The use of intelligent caching can significantly reduce the cost of your storage infrastructure. All NetApp storage systems immediately benefit from the NetApp caching optimizations that NetApp has developed for write caching and read caching in first-level buffer cache.

The optional NetApp Flash Cache serves as a second-level read cache that accelerates performance for a wide range of common applications and can reduce cost either by decreasing the number of disk spindles you need (saving space, electricity, and cooling) and/or by allowing you to purchase capacity-optimized disks rather than performance-optimized ones. Both the system buffer cache and Flash Cache exhibit a significant cache amplification effect when used in conjunction with NetApp deduplication or FlexClone technologies, and behavior can be further optimized using the workload prioritization capabilities of Data ONTAP.

NetApp FlexCache provides a third-level read cache that creates a separate caching tier within your storage infrastructure, allowing you to scale read performance beyond the capabilities of any single storage system. When you implement a caching tier, it often makes sense to concentrate investments in Flash Cache, high-performance disks, or SSDs in the caching tier, allowing you to economize on back-end storage. FlexCache can be deployed anywhere you would otherwise require a fan-out storage configuration and is popular for

use with engineering applications such as electronic design automation, animation rendering, oil and gas exploration applications, and software development. It can also accelerate data access from remote offices.

Because intelligent caching technologies immediately adapt to changes in workload, they can eliminate much of the time-consuming and complicated manual load balancing and performance tuning required with traditional storage approaches. As a result, intelligent caching can help reduce both capital and operating expenses.

9 REFERENCES

This section includes all the hyperlinks referenced throughout this document.

NetApp TR-3001: A Storage Networking Appliance: <http://media.netapp.com/documents/tr-3001.pdf>

NetApp TR-3298: RAID-DP: <http://media.netapp.com/documents/tr-3298.pdf>

NetApp WP-7005: RAID-DP, Protection Without Compromise: <http://media.netapp.com/documents/wp-7005.pdf>

Blog: Finding a Pair of Socks (Read-Ahead Algorithms):
<http://blogs.netapp.com/shadeofblue/2008/10/finding-a-pair.html>

NetApp TR-3832: Flash Cache and PAM Best Practices Guide: <http://media.netapp.com/documents/tr-3832.pdf>

NetApp TR-3669: FlexCache Caching Architecture: <http://media.netapp.com/documents/tr-3669.pdf>

Blog: Transparent Storage Cache Sharing: <http://blogs.netapp.com/virtualstoragequy/2010/03/transparent-storage-cache-sharing-part-1-an-introduction.html>

White paper: Application Mobility Across Data Centers: <http://media.netapp.com/documents/wp-app-mobility.pdf>

NetApp WP-7082: Using Flash Cache with OLTP: <http://media.netapp.com/documents/wp-7082.pdf>

NetApp DS-2838: Development and Test Solutions: <http://media.netapp.com/documents/ds-2838.pdf>

Bringing *Avatar* to Life with FlexCache: <http://communities.netapp.com/docs/DOC-6161>

10 ACKNOWLEDGMENTS

The editor would like to recognize a number of people who provided significant assistance with the development of this paper: Ingo Fuchs, Chris Gebhardt, Ashish Gupta, Alex McDonald, Steve Schuettinger, Amit Shah, Graham Smith, Vaughn Stewart, Marty Turner, Paul Updike, and Michael Zhang. Thank you for your time, ideas, content contributions, and other forms of assistance.

NetApp provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.



www.netapp.com

© 2010 NetApp. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. NetApp, the NetApp logo, Go further, faster, Data ONTAP, FlexCache, FlexClone, FlexShare, RAID-DP, Snapshot, and WAFL are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. Microsoft and Windows are registered trademarks of Microsoft Corporation. VMware is a registered trademark and vCenter is a trademark of VMware, Inc. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.
WP-7107-0810