



Technical Report

Flash Pool Design and Implementation Guide

Paul Updike, NetApp
May 2012 | TR-4070

Overview

This Technical Report covers the design considerations and best-practice recommendations for NetApp Flash Pool.

TABLE OF CONTENTS

1	What Is a Flash Pool?	4
1.1	About WAFL Blocks	4
2	How Flash Pool Works	4
2.1	Read Caching	4
2.2	Read Eviction Mechanism	5
2.3	Write Caching	5
2.4	Write Eviction Mechanism	7
3	Creating a Flash Pool	8
3.1	7-Mode CLI	8
3.2	Cluster CLI	8
4	Determining the Flash Pool Insertion Policy for a Volume	9
4.1	Insertion Policies	9
4.2	Read Caching Policies	10
4.3	Write Caching Policies	12
4.4	Flash Pool Policy Changes in Data ONTAP Operating in Cluster-Mode	12
5	Flash Pool SSD, RAID, and WAFL Design Considerations	12
5.1	Minimum Number of SSDs When Creating a Flash Pool	12
5.2	Adding SSDs to an Existing Flash Pool	13
5.3	SSD RAID Group RAID Type	13
5.4	Aggregate Snapshots	13
5.5	SAS Shelf and Storage Stack Considerations	13
5.6	Inappropriate Environments for Flash Pool	14
5.7	Flash Cache and Flash Pool Interaction	14
6	Nondisruptive Performance Expectations	14
7	Flash Pool and Predictive Cache Statistics	15
8	Performance Analysis	15
8.1	Flash Pool Read Operations	16
9	Performance Analysis of Data ONTAP Operating in Cluster-Mode	18
10	Frequently Asked Questions About Flash Pool	19

LIST OF TABLES

Table 1) <code>stats show -p hybrid_aggr output</code>	18
Table 2) Flash Pool RAID groups created with 100GB SSDs.	21

LIST OF FIGURES

Figure 1) Read Cache temperature map, insertion, access, and eviction scanner passes.	5
Figure 2) Write cache temperature map insertion, access, and eviction.....	7
Figure 3) Metadata is data that describes, locates, or references other data	10
Figure 4) Random-read policy encompasses random reads and metadata.	11
Figure 5) Cache performance recovered in less than a minute after takeover.	15

1 What Is a Flash Pool?

A Flash Pool is the newest addition to the NetApp® Virtual Storage Tier. It is a technology that allows Flash technology in the form of solid-state disks (SSDs) and traditional hard disk drives (HDDs) to be combined to form a single Data ONTAP® aggregate. When SSD and HDD technologies are combined in a Data ONTAP aggregate, the NetApp storage system takes advantage of the latency and throughput benefits of SSD while maintaining the mass storage capacity of HDD.

A Flash Pool is built from a Data ONTAP aggregate in a two-step process, described in section 3,1, "7-Mode CLI." Essentially it is the addition of SSDs into an aggregate to provide a high-bandwidth, low-latency location that is capable of caching random reads and random overwrites. The feature does not require a license and works with any NetApp SSDs and one type of HDD per Flash Pool. That is, SSD and SAS performance drives can be combined to make a Flash Pool, or SSD and SATA capacity drives can be combined to make a Flash Pool. You cannot combine SSD, SAS, and SATA into a single Flash Pool.

As a key component of the NetApp Virtual Storage Tier, Flash Pool offers a real-time, highly efficient implementation of automated storage tiering. Fine-grain promotion of hot data elements, combined with data deduplication and thin cloning, enables optimal performance and optimal use of Flash technology.

1.1 About WAFL Blocks

NetApp storage systems run the Data ONTAP operating system, which provides the NetApp Unified Storage Architecture for accessing data from a variety of protocols. The data is stored in the WAFL® (Write Anywhere File Layout) file system. The basic unit of the WAFL file system is the WAFL block, 4KB of contiguous storage space on a disk drive. Data ONTAP uses WAFL blocks as the most granular means of storing data.

Note: This report discusses WAFL blocks being inserted into and evicted from the SSDs and HDDs. Remember that what is being discussed is the fundamental unit of data that the NetApp storage system manipulates.

2 How Flash Pool Works

2.1 Read Caching

Flash Pool read caching is directly incorporated into Data ONTAP and WAFL. When data is read from disk, it is placed into WAFL memory buffers, categorized by how it came to be read into memory. This results in randomly read data being placed in a category that separates it from other reads, such as certain types of system work or sequential reads. In a Flash Pool, when data is ejected from system memory, it is evaluated based on a combination of what buffer type it is coming from and the cache insertion policy for the volume where the data exists. If it is random, and the insertion policy allows the data to be inserted into the buffer, it is then moved into the consistency point (CP) for the next write and is written to an SSD RAID group as part of the CP.

When a read operation comes into the system, it is first checked for residence in memory buffers. When it's in memory, it is directly served to the host at low latency. If the requested data is not currently in memory, a check is made to the Flash Pool cache map to see if the data exists on the SSDs in the Flash Pool. If it does, the read operation for that data is executed and the read is brought into memory and sent to the host. If the requested data is not in memory or in the Flash Pool, it is read from the HDDs. This causes it to be placed in the appropriate category of buffer and the whole process of insertion into the Flash Pool starts again. If the request was a random read operation, the data may end up cached in the Flash Pool. For more information please see NetApp Technical Report TR-3001, A Storage Networking Appliance.

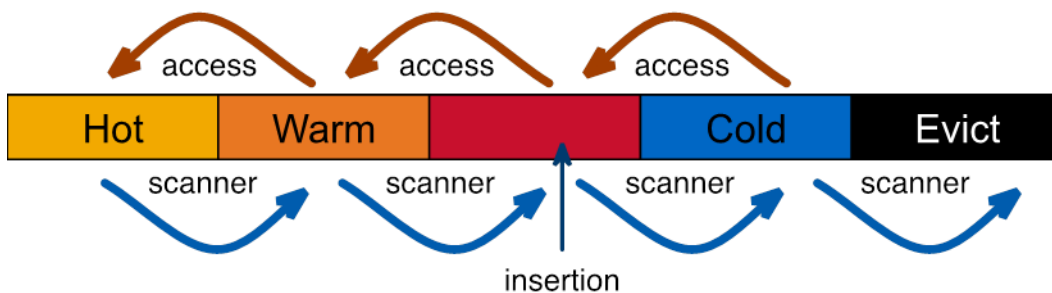
2.2 Read Eviction Mechanism

If there were an endless supply of SSD at low cost, there would be no need for Flash Pool technology. SSD would replace HDD in most environments. In reality, SSD is still considerably more expensive than HDD when compared on a cost per gigabyte of storage basis. Therefore it's important to choose carefully which data is to be inserted and which data must be evicted to make space for more data. Flash Pool keeps track of the activity of a read cached, 4KB WAFL block and evicts based on the reuse, or temperature, of the block. In this way, a more frequently accessed "hot" block is retained in the Flash Pool, while less frequently accessed "cold" blocks are selected for eviction.

When a block from a read operation is initially inserted into the Flash Pool, it is assigned a neutral value in a temperature map of the SSD cache. If it is accessed, its temperature increases from neutral to warm to hot. Hotter blocks that are accessed more frequently are tracked, while colder blocks with less frequent access are not. The eviction process doesn't come into play until the Flash Pool reaches a near full state and needs to create space to cache incoming data. At that point an eviction scanner is enabled and the most frequently accessed blocks are retained while less frequently accessed blocks are evicted from the cache.

The eviction scanner decrements the temperature of each block in the Flash Pool by one. For example, if a block was warm before the scanner ran, it would be reduced to neutral after the first eviction scanner pass. If the block was not accessed, the value would be reduced to cold on the next pass. When the block is cold and the scanner runs again, the block is evicted to make space for more data. If the block is accessed in the time between scanner passes, it is incremented. The more accesses, the more it is incremented, up to the maximum of hot. In this fashion, back and forth, the Flash Pool removes blocks that are inactive while retaining active blocks.

Figure 1) Read Cache temperature map, insertion, access, and eviction scanner passes.



2.3 Write Caching

To understand Flash Pool writes, you must first know a little about how Data ONTAP and WAFL write data to disk. An inbound write operation is stripped to its essential information and held in a nonvolatile memory location as soon as it reaches the storage system. At the same time, the write operation is processed into what will eventually be WAFL blocks on hard disk drives (HDDs). WAFL is closely integrated with NetApp RAID technology to write new data in unallocated locations. This core capability is the foundation for NetApp Snapshot™ technology and for many of the storage efficiency features that make the product a compelling solution.

The commitment of the data to NVRAM immediately, before a disk write, allows the NetApp system to maintain low latency in its response to writes. In place of a large battery-backed, RAID-mapped memory for write caches, the NetApp system uses smaller NVRAM sizes. The NVRAM is accessed only in the event of a failure; it is strictly a backup of recent write operations. All of the actual write work occurs in regular system memory. The fact that the data is written to an unused location also means that overwrites don't occur, and that most of the RAID overhead from recalculating an overwrite does not occur either. In its place, a simpler calculation of parity of the new data can occur because the writes often go to locations

of known parity – they're zeroed out. Also, all writes are treated as part of a larger write, even though they may be only a tiny change of part of a block.

NVRAM is split into two buffers for a storage system. One fills up while the other one is empty, or while its protected write operations are being written from system memory to disk. It is a little more complicated than this on a high-availability (HA) pair, but the same basic functionality applies. For more information see NetApp Technical Report TR-3450, High-Availability Pair Controller Configuration Overview and Best Practices.

All write operations are grouped into a consistency point (CP), a collection of data that is written to disk at the same time. A CP occurs for a variety of reasons that fall into three main categories: one of the NVRAM buffers is full; some system resource is too full and needs a CP to occur; or a system process requests a CP as part of its work. The `sysstat` command documentation lists a number of situations, all of which fall into these basic categories.

An important point about the combination of NVRAM, WAFL, and CP operations is the fact that NetApp storage systems are optimized for writes. In most workloads, the system is able to satisfy a write request in the time it takes NVRAM to commit the data, which is less than a millisecond. All of the other operations occur asynchronously from the operation.

When a system doesn't have enough disk drives to accept the write workload, writes may have to start running synchronously with disk operations. This means that instead of RAM speeds, the system has to write at disk speeds.

This is not a limitation of NVRAM; larger NVRAM would not improve it. More NVRAM and cached data would only gap the amount of time before the flood of write requests overflows NVRAM. The limitation in these scenarios is actually in the bandwidth, in IOPS or in bytes/sec, provided by the HDDs. If the disks aren't able to ingest that data rapidly enough, this situation may occur. Appropriate sizing of the number of disks to the workload provides enough bandwidth to the system. This can lead to sizing more disks for the workload than necessary, from a capacity point of view. That is, the number of disks is sized based on performance so that they provide enough I/O capacity to handle the work.

In the same way, read caching is designed to offload random reads from HDD and replace those operations with reads from SSD; write caching is designed to offload random overwrites from HDD and replace those with writes of blocks to SSD. Flash Pool specifically targets random overwrites because they are smaller and IOPS intensive—the perfect candidate for a caching technology.

Flash Pool does not attempt to capture sequential writes; because they are larger, they would require enormous caches, and HDD can handle sequential work well when measured on a time per byte (latency) basis. When the HDDs can handle the work efficiently, it is much more cost effective to use them. Likewise, sequential writes don't tend to be overwritten, so the disk I/O replaced would be almost nonexistent, while the data would eventually write the same amount of operations when going to HDD anyway. Finally, the biggest differences between SSD and HDD performance are in the areas of random operations, again reinforcing the caching of random data in the Flash Pool.

Flash Pool does not replace the use of NVRAM as the primary write optimization. It provides a separate location for random overwrites to be written. This can reduce the number of HDDs required for a workload and offers greater benefit to the existing write optimizations of Data ONTAP.

For write data to be cached, two criteria must be met:

- The write operation is to a random location as generated by the host (not part of a stream of sequential writes)
- The previous write operation to this same block was random as well

When these criteria are met, a write is designated for the SSDs in a Flash Pool rather than to HDD, and in the CP the data is sent to SSD. Other writes that do not meet those criteria are still sent to HDD. The data in Flash Pool write caching, unlike read caching, exists only on the SSDs. By using SSD for random

overwrites, the system can offload the hard disk drives for other work, reduce the required number of HDDs for a workload, and potentially improve latency in some workloads.

Benefits of Flash Pool Write Cache

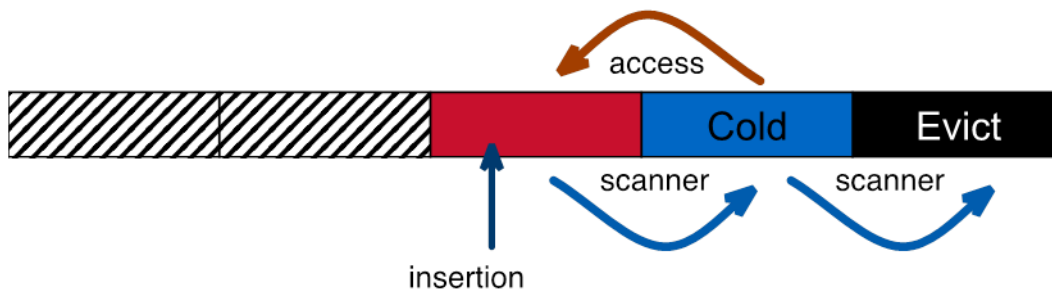
Here are the benefits of Flash Pool write caching:

- Get more out of HDD. By absorbing costly I/O that would have gone to HDD, the drives are open for more reads or writes. This translates into business value in the following ways:
 - Change from performance HDDs to capacity HDDs (for example, SAS to SATA)
 - Reduce drive counts to satisfy workload requirements
- Provide a prepopulated read cache. The data in the Flash Pool write cache is also available for reads, further reducing HDD operations.

2.4 Write Eviction Mechanism

Like reads, write cache overwrite operations are inserted into the SSDs with a neutral temperature. Unlike reads, the only way the temperature of a block can change is when the eviction scanner is running, at which point the scanner decrements the block by one level. If the block isn't overwritten after two eviction scanner passes, it is removed. If the block is overwritten between eviction scanner passes, it is reassigned the neutral value and the scanner decrements from that point. In this fashion, hot writes are constantly adjusted back to a higher temperature, while cold blocks are aged out of the cache. When a block is eventually aged out of the SSDs, it is reread into main memory as part of the next CP and written to disk at the time of that CP.

Figure 2) Write cache temperature map insertion, access, and eviction.



Flash Pool and Storage Efficiency

Storage efficiency features, deduplication, and thin clones are maintained in the Flash Pool. Just as with most data at rest in the aggregate, data in the Flash Pool cache is maintained as 4KB WAFL blocks. Compressed data, however, is maintained in a different fashion on disk; as a result, compressed data cannot be cached in Flash Pool. Blocks that aren't compressed but that are in a compressed volume can be cached; only the compressed data cannot be cached.

For deduplication and thin clones, this has an effect called *cache amplification*. Although the amount of physical Flash in the SSDs is fixed, the amount of data represented by those SSDs can potentially be far larger. This is because the same HDD storage efficiency features make the Flash Pool SSD cache more efficient as well.

3 Creating a Flash Pool

3.1 7-Mode CLI

A Flash Pool is based on the concept of an aggregate in Data ONTAP. From a system running Data ONTAP 8.1.1 or later, a 64-bit aggregate can be upgraded to a Flash Pool by following a simple two-step procedure.

1. The aggregate must be enabled to accept SSDs to become a Flash Pool.
2. SSDs are added to the aggregate in a new RAID group.

The `aggr create` command is used to create an aggregate as usual.

```
FAS3240-ONTAP-FP> aggr create aggr1_FP -T SAS -t raid_dp -B 64 16@560
```

The `aggr options` command is used to set the `hybrid_enable=on` option, which enables the aggregate to be converted into a Flash Pool.

```
FAS3240-ONTAP-FP> aggr options aggr1_FP hybrid_enabled on
```

Adding SSDs to the aggregate in a new SSD-only raid group is accomplished with the `aggr add` command, specifying SSD as the disk type to add.

```
FAS3240-ONTAP-FP> aggr add aggr1_FP -T SSD 6@100
```

The status of the Flash Pool can now be shown with the `aggr status -v` command. Note that the `hybrid_enabled` option is now on.

```
FAS3240-ONTAP-FP> aggr status -v
Aggr State      Status      Options
aggr1_FP online  raid_dp, aggr  nosnap=off, raidtype=raid_dp, raidsize=16,
64-bit         hybrid       ignore_inconsistent=off, snapmirrored=off,
               resyncsnaptime=60, fs_size_fixed=off,
               snapshot_autodelete=on, lost_write_protect=on,
               ha_policy=cfo, hybrid_enabled=on,
               percent_snapshot_space=0%,
               free_space_realloc=off
```

3.2 Cluster CLI

The basic process followed for Data ONTAP operating in Cluster-Mode is the same as in the 7-Mode CLI. That is, set an option to allow the aggregate to become a Flash Pool and then add SSDs in a new RAID group. When these two steps are complete, the aggregate is a Flash Pool.

The `storage aggregate` command is used to create an aggregate as usual. The `storage aggregate -modify` command is used to set a `-hybrid_enabled true` option, which enables the aggregate to become a Flash Pool.

```
FAS3240-ONTAP-FP> storage aggregate modify -aggregate aggr1 -hybrid-enabled true
```

SSD disks are added to an SSD-only RAID group in the Flash Pool by using the `storage aggregate add-disks` command.

```
FAS3240-ONTAP-FP> storage aggregate add-disks -aggregate aggr1 -disktype SSD -diskcount 3
```

The status of the newly converted aggregated can be confirmed with the `storage aggregate show` command.

```
FAS3240-ONTAP-FP> storage aggregate show -aggregate aggr1
```



```

Aggregate: aggr1
Checksum Style: block
Number Of Disks: 6
Mirror: false
Nodes: raid-u87
Disks for First Plex: raid-u87:6a.25.16,
raid-u87:6b.25.17,
raid-u87:6a.25.18,
raid-u87:6b.25.13,
raid-u87:6a.25.14,
raid-u87:6b.25.15
Disks for Mirrored Plex: -
Free Space Reallocation: off
HA Policy: sfo
Space Reserved for Snapshot Copies: -
Hybrid Enabled: true
Available Size: 744.9GB
Block Type: 64-bit
Checksum Enabled: true
Checksum Status: active
DR Home ID: -
DR Home Name: -
Has Mroot Volume: false
Has Partner Node Mroot Volume: false
Home ID: 1873775299
Home Name: raid-u87
Total Hybrid Cache Size: 82.57GB
Hybrid: true
Inconsistent: false
Is Aggregate Home: true
Max RAID Size: 14
Hybrid Aggregate SSD Tier Maximum RAID Group Size: 23
Owner ID: 1873775299
Owner Name: raid-u87
Used Percentage: 0%
Plexes: /aggr1/plex0
RAID Groups: /aggr1/plex0/rg0 (block)
/aggr1/plex0/rg1 (block)
RAID Status: raid_dp, hybrid, normal
RAID Type: raid_dp
SyncMirror Resync Snapshot Frequency in Minutes: 60
Is Root: false
Space Used By metadata for Volume Efficiency: 0B
Size: 744.9GB
State: online
Used Size: 180KB
Number Of Volumes: 0
Volume Style: flex

```

For more information about Flash Pool administration, see the Data ONTAP 8.1.1 product documentation.

4 Determining the Flash Pool Insertion Policy for a Volume

Flash Pool can be configured on a volume-by-volume basis. The policy controls what data can be inserted into the volume. The policy does not control the residence time of the data once it is cached or change the eviction behavior when it is time to free space for more data. No priorities for cached data are applied at the time of insertion; all data is inserted with the same priority. The policy's sole purpose is to allow a yes/no decision for a block of data to determine whether it is eligible to be cached.

4.1 Insertion Policies

The mechanism that controls which blocks are cached or not cached is the Flash Pool insertion policy. There are separate policies for reads and writes for each volume, because the two types of data are managed differently in Flash Pool. In the majority of workloads, NetApp recommends retaining the

defaults. Policy changes take effect immediately and change the behavior of data being inserted into the volume. However, data that is already in the SSD remains until it is aged out according to the eviction mechanisms described in section 2.2, "Read Eviction Mechanism" and section 2.4, "Write Eviction Mechanism."

4.2 Read Caching Policies

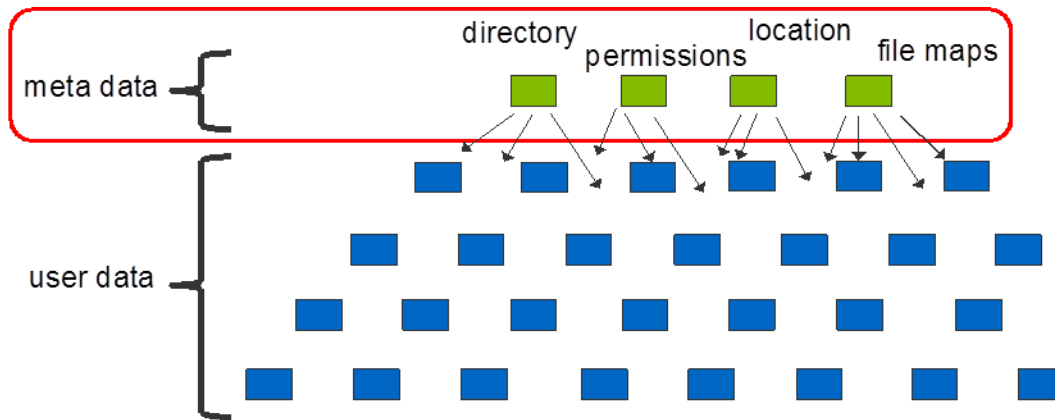
Read cache policies affect the insertion of data that is allowed in the cache from read operations. This behavior is based only on eligible data; sequential reads and certain system work are not made eligible even through the insertion policies.

Note: Changing the read and write cache policy requires a change in the command line privilege level to Advanced. This mode contains commands that could potentially cause an outage or a loss of data. For that reason, when changing a Flash Pool insertion policy it is best to enter Advanced mode as shown in the examples, change the policy, and immediately set the privilege level back to Admin.

read-cache=meta

The metadata-only policy allows the metadata only from the volume that has this option set. In most scenarios, user data should be cached as well as the metadata. In some situations; especially in very large datasets with uniform random access across most of the data, metadata-only mode may be more beneficial than the default because it is unlikely to get a read cache hit on user data, whereas metadata is used. In addition, the amount of metadata in very large random workloads can be a significant portion of the overall reads.

Figure 3) Metadata is data that describes, locates, or references other data



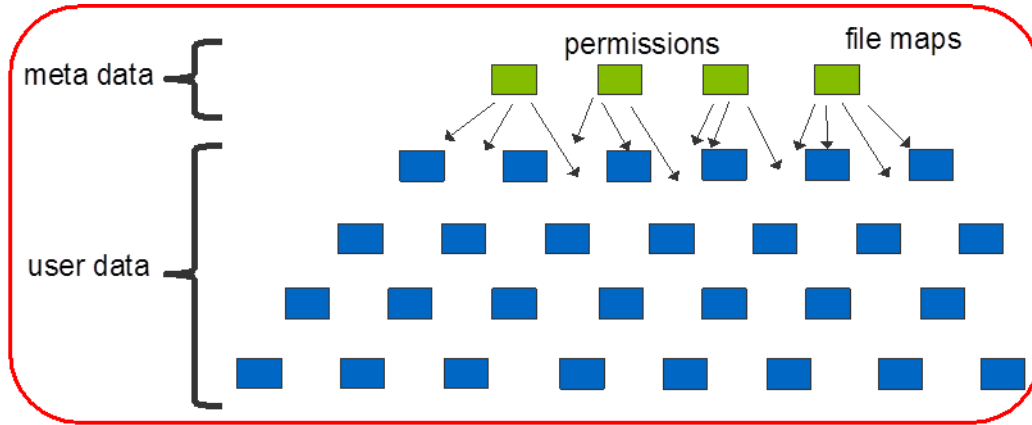
Metadata is defined as buffers associated with indirect blocks—blocks that point to other blocks, directory blocks for NAS file systems, and various map files and information used by Data ONTAP and WAFL to keep track of user data.

```
FAS3240-ONTAP-FP> priv set advanced
Warning: These advanced commands are potentially dangerous; use
        them only when directed to do so by NetApp
        personnel.
FAS3240-ONTAP-FP*> priority hybrid-cache set voll read-cache=meta
FAS3240-ONTAP-FP*> Sun May 6 14:28:51 EST [FAS3240-ONTAP-FP:waf1.hya.rcache.policy:info]: Read
caching policy on volume 'voll' has been modified to 'meta'.
FAS3240-ONTAP-FP> priv set admin
```

read-cache=random-read

The random read policy is the default policy and should rarely be changed. When this policy is set, both user data and metadata are placed into the Flash Pool SSDs for that volume. Random reads are described in section 2.1, "Read Caching."

Figure 4) Random-read policy encompasses random reads and metadata.



```
FAS3240-ONTAP-FP> priv set advanced
Warning: These advanced commands are potentially dangerous; use
        them only when directed to do so by NetApp
        personnel.
FAS3240-ONTAP-FP*> priority hybrid-cache set voll read-cache=random-read
FAS3240-ONTAP-FP*> Sun May 6 14:25:02 EST [FAS3240-ONTAP-FP:waf1.hya.rcache.policy:info]: Read
caching policy on volume 'voll' has been modified to 'random-read'.
FAS3240-ONTAP-FP> priv set admin
```

read-cache=random-read-write

This is a read caching policy, not a write caching policy. For random-read-write, random read and random write blocks are written into the read cache and are immediately available for reads. This allows random writes, not just overwrites. When this option is enabled, blocks can be duplicated in the read cache and the write cache. This can help some workloads with a high ratio of read-to-write rates when it is desirable to avoid going to disk for any read operations. This policy does not result in write-offload from the HDDs for write operations and therefore is useful only in very specific situations.

```
FAS3240-ONTAP-FP> priv set advanced
Warning: These advanced commands are potentially dangerous; use
        them only when directed to do so by NetApp
        personnel.
FAS3240-ONTAP-FP*> priority hybrid-cache set voll read-cache=random-read-write
FAS3240-ONTAP-FP*> Sun May 6 14:00:25 EST [FAS3240-ONTAP-FP:waf1.hya.rcache.policy:info]: Read
caching policy on volume 'voll' has been modified to 'random-read-write'.
FAS3240-ONTAP-FP> priv set admin
```

read-cache=none

When this policy is set for a volume, data from that volume is not available to be cached in the Flash Pool.

```
FAS3240-ONTAP-FP> priv set advanced
Warning: These advanced commands are potentially dangerous; use
        them only when directed to do so by NetApp
        personnel.
FAS3240-ONTAP-FP*> priority hybrid-cache set voll read-cache=none
```

```
FAS3240-ONTAP-FP*> Sun May 6 14:31:15 EST [FAS3240-ONTAP-FP:waf1.hya.rcache.policy:info]: Read
caching policy on volume 'voll' has been modified to 'none'.
FAS3240-ONTAP-FP> priv set admin
```

4.3 Write Caching Policies

random-write

This is the default policy. When set, the write caching behavior described in section 2.3 is enabled. This policy should be applied in a majority of environments to enable HDD offload for random overwrites.

```
FAS3240-ONTAP-FP> priv set advanced
Warning: These advanced commands are potentially dangerous; use
        them only when directed to do so by NetApp
        personnel.
FAS3240-ONTAP-FP*> priority hybrid-cache set voll write-cache=random-write
FAS3240-ONTAP-FP*> priority hybrid-cache set voll write-cache=random-write
FAS3240-ONTAP-FP*> Sun May 6 14:49:08 EST [FAS3240-ONTAP-FP:waf1.hya.wcache.policy:info]: Write
caching policy on volume 'voll' has been modified to 'random-write'.
FAS3240-ONTAP-FP> priv set admin
```

none

When this option is set, write caching for the volume is disabled.

```
FAS3240-ONTAP-FP> priv set advanced
Warning: These advanced commands are potentially dangerous; use
        them only when directed to do so by NetApp
        personnel.
FAS3240-ONTAP-FP*> priority hybrid-cache set voll write-cache=none
FAS3240-ONTAP-FP*> Sun May 6 14:48:43 EST [FAS3240-ONTAP-FP:waf1.hya.wcache.policy:info]: Write
caching policy on volume 'voll' has been modified to 'none'.
FAS3240-ONTAP-FP> priv set admin
```

4.4 Flash Pool Policy Changes in Data ONTAP Operating in Cluster-Mode

The Flash Pool policies are available at the node shell when operating in Cluster-Mode. To enter the node shell and then set the insertion policy, follow this procedure:

```
Cluster-01>run -node -node01
node01> priv set advanced
Warning: These advanced commands are potentially dangerous; use
        them only when directed to do so by NetApp
        personnel.
node01*> priority hybrid-cache set voll write-cache=none
node01*> Sun May 6 14:48:43 EST [FAS3240-ONTAP-FP:waf1.hya.wcache.policy:info]: Write caching
policy on volume 'voll' has been modified to 'none'.
node01> priv set admin
```

Once in the node shell, the insertion policies can be set in the same fashion as in the 7-Mode, using the `priority` command to change the hybrid-cache setting for the volume in question.

5 Flash Pool SSD, RAID, and WAFL Design Considerations

5.1 Minimum Number of SSDs When Creating a Flash Pool

To maximize the performance of a NetApp system with Flash Pool, a minimum number of solid state disks should be used in each aggregate that is being converted to Flash Pool. This is based on the amount of bandwidth available from the cache SSDs versus the system's performance capability. In each Flash Pool, NetApp recommends using the minimum number of SSDs required to match the system's

ability. If there are multiple Flash Pool aggregates in the system, each aggregate should be built with the minimum number of disks.

Use the following guideline when creating a Flash Pool with SSDs added to an aggregate:

- For FAS2000 series that support Flash Pool, there is no required minimum number of SSDs.
- For FAS3100 and FAS3200 series that support Flash Pool, the recommended minimum number of SSDs is a 3+2 (3 data + 2 parity) RAID group. (This assumes NetApp RAID-DP[®] protection.)
- For FAS6000 and FAS6200 series that support Flash, the required minimum number of SSD data drives is 9+2 (9 data + 2 parity) RAID group. (This assumes RAID-DP protection.)

If this recommendation is not met, a bottleneck could be created in the cache, artificially limiting system performance.

5.2 Adding SSDs to an Existing Flash Pool

NetApp also recommends adding drives to the Flash Pool and growing the size of the SSD cache. These recommendations are based on the write patterns after adding new drives. When drives are added to a Flash Pool, they are empty and Data ONTAP starts writing to them until they are leveled with the rest of the drives in the RAID group. For this reason, drives should be added in increments so that the additional drives do not become a system bottleneck.

Use the following guidelines when adding SSDs to an existing Flash Pool:

- For FAS2000 series that support Flash Pool, there is no required minimum number of SSDs when adding drives.
- For FAS3100 and FAS3200 series that support Flash Pool, the minimum recommended number of SSDs to add at a time is three. That is, no fewer than three data drives should be added at a time, excluding parity drives, in each Flash Pool.
- For FAS6000 and FAS6200 series that support Flash Pool, the minimum recommended number of SSDs to add at a time is six. That is, no fewer than six data drives should be added at a time, excluding parity drives, in each Flash Pool.

If a new RAID group has to be created, these recommendations still hold, but the requisite number of drives must also include parity drives.

5.3 SSD RAID Group RAID Type

The SSD RAID group in each Flash Pool has the same RAID protection as the aggregate where it was created. For example, creating a Flash Pool from an aggregate with RAID-DP protection requires the SSD RAID group to be protected by RAID-DP as well. If the aggregate is RAID 4, the SSD in the Flash Pool is also RAID-4.

5.4 Aggregate Snapshots

Aggregate Snapshot copies should be disabled and existing aggregate Snapshot copies should be deleted before enabling Flash Pool on an existing aggregate. Aggregate Snapshots copies should not be enabled on a newly created Flash Pool.

5.5 SAS Shelf and Storage Stack Considerations

The SSDs in a Flash Pool are capable of very high IOPS and bandwidth. This creates an important design consideration when introducing a shelf with SSDs into a NetApp storage system to create a Flash Pool.

- In mixed SSD and HDD stacks:
 - FAS2xxx: 4 shelves (6-12 SSDs) per stack

- 2240: 3 external (+1 internal) shelves (6-12 SSDs) per stack
- 2220: 2 external (+1 internal) shelves (6-12 SSDs) per stack
- FAS3xxx: 2-4 shelves (12-24 SSDs) per stack
- FAS6xxx: 2 shelves (12 SSDs) per stack
- Full SSD shelves (24 SSDs) are best placed in their own stack
- Expand with SSDs as the first shelf in stack

Stack means the stack of disk shelves cabled into a SAS I/O port on a storage system, as defined in the installation guides for the DS4243 and DS2246 shelves.

5.6 Inappropriate Environments for Flash Pool

- Flash Pool does not help with systems that are CPU saturated or controller or node limited. It cannot produce greater performance on such a system and may cause a slight degradation when added to such a system.
- Environments that are mostly sequential reads are not likely to benefit a great deal from Flash Pool.
- Environments that are mostly sequential writes and not likely to benefit a great deal from Flash Pool.

5.7 Flash Cache and Flash Pool Interaction

NetApp Flash Cache is a controller-based Flash technology. It can cache data for all workloads in the system, unlike NetApp Flash Pool, which is specific to an aggregate. Because it is in the controller, Flash Cache cannot transmit its cache contents in the event of a storage takeover. In that case, Flash Pool moves the SSD cache to the other node and continues operations.

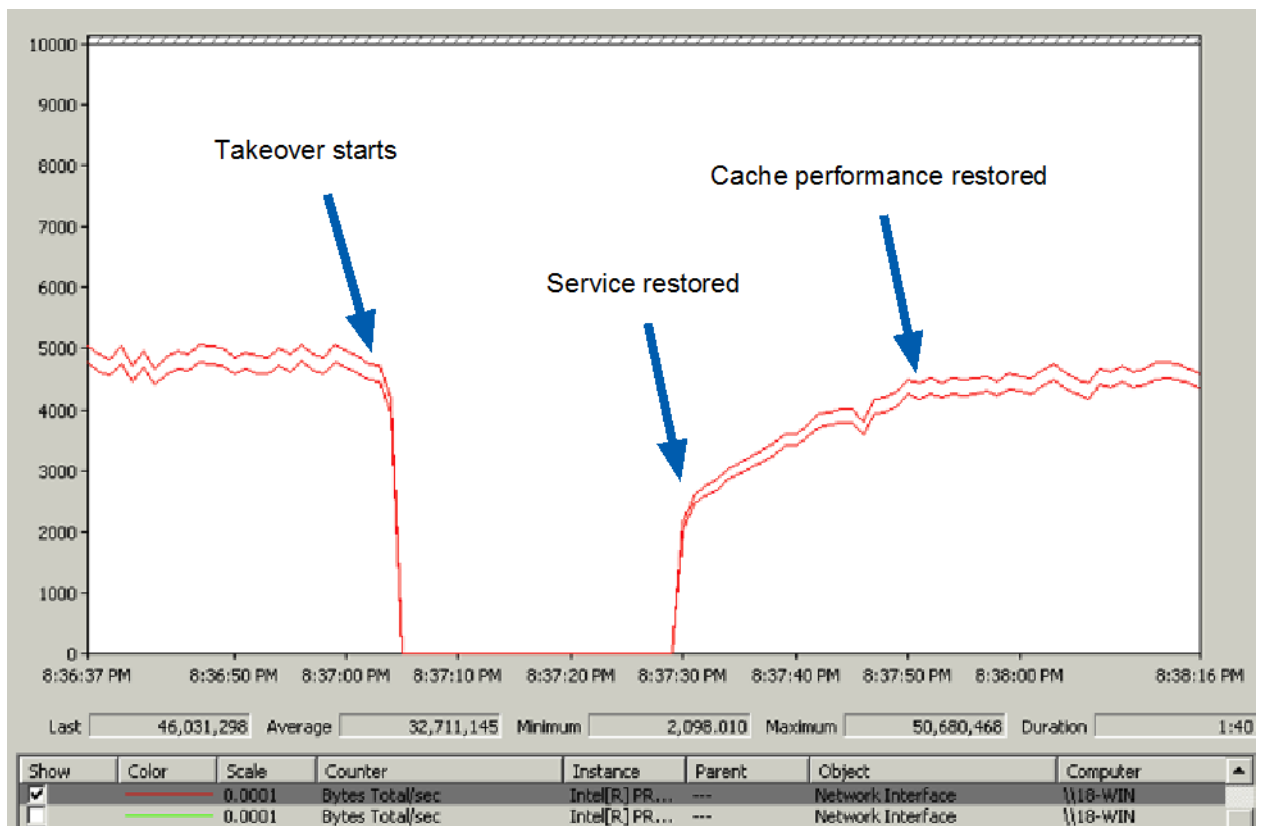
When Flash Cache and Flash Pool are used in the same system, Flash Cache does not cache data from a Flash Pool aggregate. There is no benefit in caching the data a second time, so Flash Cache ignores that data. If Flash Cache is installed in a system with Flash Pool and is not being used, it should be removed from the system to enable better performance from the Flash Pool.

6 Nondisruptive Performance Expectations

Flash Pool is physically implemented in the storage subsystem. As a result, both nodes in an HA pair have access to the SSD drives in case of a takeover of one node by the other. This may occur due to a planned takeover, as in an upgrade, or an unplanned failure event on the node. Because the cache structures are in SSD, a takeover does not affect the read and write cached data or its accessibility once running in takeover mode. Any data in the SSD portion of the Flash Pool continues to be served at low latency from the SSD after the event. Likewise, when the node is available for giveback, the Flash Pool is available immediately. If both sides of an HA pair are powered off, the Flash Pool retains the cached contents in SSD and is available on reboot.

Figure 5 shows three phases in the workload: before the takeover event, during the takeover, and after the takeover. A load generator drove this workload, driving random read and write work to the system, much of which was served from the SSD. As shown in the graph, the workload recovers after the takeover and continues operating at the same throughput and latency as before.

Figure 5) Cache performance recovered in less than a minute after takeover.



7 Flash Pool and Predictive Cache Statistics

Predictive Cache Statistics (PCS) help decide whether or not Flash Cache read caching would benefit a system workload. PCS can be used with Flash Pool read caching as well, under the following guidelines:

- PCS does not work if Flash Cache or Flash Pool is already present in the system.
- Flash Pool read caching tends to see slightly better results than PCS may predict. This is mostly due to the effect of read hits from write cached. When looking at PCS, assume a slight increase in the hit rate.
- Set the PCS cache size to 75% of the intended usable SSD size in the Flash Pool. This setting takes into account the self-optimization that the Flash Pool cache does to make sure that it can absorb bursty workloads.
- Takes into account read caching. Be conservative and adjust the results down by 10% to 20%, depending on the write traffic.

Predictive Cache Statistics functionality is documented in [TR-3801: PCS Best Practices Guide](#).

8 Performance Analysis

Performance analysis of a system that includes Flash Pool is similar to analysis of a system without Flash Pool. The Flash Pool functionality should be viewed as an extension of a NetApp system and Data ONTAP. Therefore the same concepts apply when investigating the performance of the system. The focus of any storage system analysis is centered on the average latency for the protocol operations. With an aggregate that has been upgraded to a Flash Pool, the considerations are whether data is being cached; whether that cached data is being hit on later accesses; and the latency of the system.

The Flash Pool can be separated into its read behaviors and its write behaviors, as discussed in section 2, "How Flash Pool Works." Due to the different nature of what a "hit" is in the read cache versus the write cache, the two are covered separately.

8.1 Flash Pool Read Operations

Flash Pool's effectiveness in read operations is measured by the amount of reads it is replacing. This can be further broken down into the WAFL blocks that are being replaced versus the actual disk operations replaced. The Flash Pool cache is designed to offload HDD operations to SSDs. In this respect, it's important to understand the actual number of disk reads that don't have to be serviced by HDD. WAFL blocks are 4KB, and disk operations occur in multiples of the 4KB WAFL block. For example, an 8KB operation is satisfied by two WAFL blocks most of the time. For this reason, the number of operations replaced may vary in respect to the WAFL blocks hit in the cache. A ratio with fewer I/Os replaced to WAFL blocks indicates relatively larger disk operations. A ratio that approaches 1:1, where the disk I/Os are much smaller, has more relative effect on the system. The object is to replace more disk operations.

This is important when looking at disk throughput for SSD and HDD during a workload. Because SSDs are handling small random I/O, the amount of data coming from HDDs may still be significantly greater than that coming from SSDs. It is important to investigate the amount of HDD I/O being replaced, because HDDs are handling larger sequential operations, which they handle well compared to random read operations. Removing the random read operations means more bandwidth for sequential work from the HDD. In addition, HDDs have to use seek time for each random read, potentially costing as much as 10ms on even lightly loaded drives. The SSD is capable of handling these operations at submillisecond latencies even with a workload that would be 10 times what an HDD could accomplish at its maximum IOPS.

In the case of a highly sequential workload, not much data is cached in the SSDs, and a small number of I/Os are replaced. However, even in these workloads, metadata operations—which are small random reads—are cached. This allows faster access to metadata and faster response times once the data is looked up and comes from the HDD. A random read workload with a working set size (the amount of currently active data) that is significantly larger than the amount of cache space provided by the SSDs shows a smaller number of reads replaced as well, but it may still benefit from metadata hits.

There is one main command to focus on when looking into Flash Pool performance: the `stats show` command, with a preset output for displaying Flash Pool statistics.

```
FAS3240-ONTAP-FP> stats show -p hybrid_aggr
```

The `stats show` command can also be used with the protocol of interest to see the performance at the data object in question. This example shows the LUN latency.

```
FAS3240-ONTAP-FP> stats show lun:*.avg_latency
```


From the output of the `stats show` command, note that both HDD and SSD are seeing read operations coming from them. From the simple `sysstat` output, it's hard to tell how effective the reads from SSD are in replacing disk operations, because `sysstat` reports KB/sec for disks. To determine the relative effectiveness of the SSD read cache, the `stats show` command, above, can be used:

```
FAS3240-ONTAP-FP> stats show -p hybrid_aggr
```

Instance	ssd used	read cached	write cached	read ops replaced	rate %	write blks replaced	rate %	rd cache evict	wr cache destage	rd cache ins rate	wr cache ins rate	read hit latency	read miss latency
aggr1_FP 25463723	22517078	997993	1262	38	0	0	0	0	0	631	0	0.64	18.48
aggr1_FP 25445886	22537910	998062	1217	38	142	0	0	0	0	5457	142	2.01	26.20
aggr1_FP 25454002	22537910	998062	1060	37	0	0	0	0	0	2915	0	0.71	17.48
aggr1_FP 25456742	22535137	997209	2750	47	0	0	0	0	0	4961	0	2.90	18.69
aggr1_FP 25465315	22534297	997209	1524	43	0	0	0	0	0	7678	0	0.94	18.33
aggr1_FP 25472547	22535385	997337	1314	42	127	0	0	0	0	5845	127	0.78	17.11
aggr1_FP 25449077	22535979	998982	3101	47	2086	3	0	0	0	5090	2086	3.02	25.57
aggr1_FP 25455710	22533723	998937	2015	46	0	0	0	0	0	9606	0	2.72	30.91
aggr1_FP 25467649	22532691	998937	1581	42	0	0	0	0	0	11500	0	0.88	17.96
aggr1_FP 25475870	22532691	998937	1436	41	0	0	0	0	0	7174	0	0.76	16.81
aggr1_FP 25481560	22536518	1001106	4593	60	2169	4	0	0	0	4292	2169	4.33	41.23
aggr1_FP 25474499	22536203	1001987	2027	45	1642	9	0	0	0	19896	1642	3.04	20.04
aggr1_FP 25483545	22534175	1001987	1543	41	0	0	0	0	0	4765	0	1.57	25.59
aggr1_FP 25492483	22534175	1001987	1246	38	0	0	0	0	0	1200	0	0.92	18.74
aggr1_FP 25501951	22554689	1002500	3052	49	512	1	0	0	0	4434	512	3.56	21.31
aggr1_FP 25484783	22554783	1002173	2034	48	2	0	0	0	0	10485	2	1.92	30.89
aggr1_FP 25493584	22554783	1002173	1356	43	0	0	0	0	0	7738	0	0.98	18.66
aggr1_FP 25503240	22554158	1002173	1480	40	0	0	0	0	0	11094	0	0.70	17.43
aggr1_FP 25513702	22556590	1002365	1472	43	192	1	0	0	0	10919	192	1.23	19.07
aggr1_FP 25504000	22566363	1005567	1955	49	4120	10	0	0	0	13652	4120	2.68	29.17
aggr1_FP 25510240	22563403	1005255	2488	46	0	0	0	0	0	4757	0	2.54	15.96
aggr1_FP 25526127	22563305	1005255	1538	43	0	0	0	0	0	13479	0	0.98	18.80
aggr1_FP 25515226	22583433	1006978	2618	49	1723	4	0	0	0	8171	1723	2.39	32.08

The output is generated every 5 seconds by default, but it conforms to the standard syntax of the `stats` command, allowing the output frequency to be changed by use of the `-i` switch. For example, `-i 1` would change the output to every second.

The statistics counters that the output reports are all on a per-second basis. By default, the output reports at five 5-second intervals. When rates are represented as per-second, they are actually reporting the per-second average over the last interval of collection. Table 1 shows how the columns are defined.

Table 1) `stats show -p hybrid_aggr` output.

Column Heading	Definition
instance	The Flash Pool instance that this row of output describes.
read cached	The amount of data currently in the read cache, in 4KB WAFL blocks.
write cached	The amount of data currently in the write cache, in 4KB WAFL blocks.
read ops replaced rate /s	The number of disk read operations replaced, per second. This is in actual disk I/Os, not WAFL blocks.
read ops replace rate %	The percentage of disk read operations replaced, on per-second interval averaged from one line of output to the next.
write blocks replaced rate /s	The number of write blocks replaced as writes to SSD instead of HDD. This is in 4KB WAFL blocks.
write blocks replaced rate %	The percentage of write blocks replaced in SSD instead of HDD, on per-second interval averaged from one line of output to the next.
rd cache evict /s	The read cache evictions per second, in 4KB WAFL blocks.
wr cache destage /s	The write cache evictions per second, in 4KB WAFL blocks.
rd cache ins rate /s	The read cache insertions per second, in 4KB WAFL blocks.
wr cache ins rate /s	The write cache insertions per second, in 4KB WAFL blocks.
read hit latency	The latency, in milliseconds, of read operations that are successfully accessed in the Flash Pool SSD cache.
read miss latency	The latency, in milliseconds, of read operations that are unsuccessfully accessed in the Flash Pool SSD and that must be served from HDD instead.

Finally, look at the individual LUN to see the performance it is receiving in this workload. Once again, use the `stats` command, this time looking only at the latency of the LUN.

```
FAS3240-ONTAP-FP> stats show lun*:avg_latency
lun:/vol/vol1/lun1-7STwF?AvYi0K:avg_latency:5.50ms
```

This process can be used on volume-level objects, on protocols (NFS, CIFS, and so on), or on a LUN level as just described. For more information about the counters that can be viewed in this way, run `stats show` to retrieve a list of counters and then run `stats explain` on the counters in question.

9 Performance Analysis of Data ONTAP Operating in Cluster-Mode

Cluster performance analysis of Flash Pool is very similar to that of 7-Mode. In both modes, the technology works in WAFL to access and store data. As a result, it is below protocol traffic and cluster-level communication. When observing the performance of Flash Pool in Data ONTAP 8 operating in Cluster-Mode, the same behavior described in section x.x applies. The only additional step is to first change to the node shell.

```
Cluster-01>run -node -node01
```

In the node shell, the `stats show -p hybrid_aggr` command can be used, as in 7- Mode. From there, the same setup steps, commands, and definitions of output described in section 6 apply.

10 Frequently Asked Questions About Flash Pool

How does Flash Pool work with SnapMirror?

Answer: Flash Pool does not transmit SSD information to a NetApp SnapMirror® or SnapVault® destination. When a destination is brought online, it has no knowledge of which contents were in the SSD or HDD, even if the destination volume is on a Flash Pool. In this scenario, the destination independently loads data into the SSDs based on the local access patterns to the volume.

What kind of disk drives can be used in a Flash Pool?

Answer: A Flash Pool can be created from NetApp SSDs and one type of spinning drive. The SSDs are placed in a separate RAID group from the spinning drives. Only one type of SSD and one type of spinning drive can be used to make a Flash Pool.

Can different disk drive types be used in a single Flash Pool?

Answer: Only one type of spinning drive can be combined with SSDs to create a Flash Pool. For example, it's acceptable to have SATA and SSD, or you can have SAS (or FC) and SSD. The combination of SATA and SAS and SSD is not allowed in the same aggregate.

Can I create a Flash Pool by using two different types of HDDs (no SSD)?

Answer: No. The relative performance gains of SAS over SATA don't provide enough benefit to make it worthwhile. A Flash Pool must contain SSD and one other disk type.

What kind of aggregate can be converted to a Flash Pool?

Answer: A Flash Pool can only be created from 64-bit aggregates. You cannot create a 32-bit Flash Pool or convert an existing 32-bit aggregate into a Flash Pool. Data ONTAP 8.1 can nondisruptively expand 32-bit aggregates to 64-bit. Once the expansion process is triggered, the aggregate can be converted to a Flash Pool.

Can read or write caching be tuned in a Flash Pool?

Answer: Yes. Read and write caching can be modified to insert data based on different workloads with volume-level settings made with the `priority hybrid-cache` command. For full information, see the Data ONTAP documentation.

Does the SSD RAID group have to be the same RAID protection level as the HDD RAID groups in a Flash Pool?

Answer: Yes. The entire aggregate must be at the same RAID protection level, either all RAID-DP technology or all RAID 4. NetApp recommends RAID-DP for general use due to its higher fault tolerance; this recommendation does not change with the addition of SSD in a Flash Pool.

Can I make a Flash Pool by using a V-Series system and other storage vendors' arrays?

Answer: No. You must use NetApp hard disk drives and disk shelves plus NetApp SSDs. You can create a Flash Pool on a V-Series system, but you must use NetApp HDDs and SSDs.

Can SSDs be removed from a Flash Pool and be reused elsewhere?

Answer: No, not without destroying the aggregate. Once the SSDs are placed in a Flash Pool they are required in that aggregate as long as it exists.

Which release of Data ONTAP supports creation of a Flash Pool?

Answer: Flash Pool is available starting in Data ONTAP 8.1.1 operating in both 7-Mode and Cluster-Mode.

Can I convert an existing aggregate into a Flash Pool?

Answer: Yes. An existing aggregate created with FCAL, SAS, or SATA drives can be converted into a Flash Pool.

In which release of Data ONTAP must an aggregate have been created before I can turn it into a Flash Pool?

Answer: In general, if an aggregate is a 64-bit aggregate it can be converted into a Flash Pool. However, there are some scenarios that restrict conversion of an aggregate to a Flash Pool. See the Data ONTAP 8.1.1 product documentation for information about those situations.

Which SSDs can be used with Flash Pool?

Answer: Any NetApp SSDs not currently used in an aggregate can be used to create a Flash Pool.

Does the capacity of SSDs count toward the total capacity of a Flash Pool?

Answer: No. For example, if an aggregate is already at its maximum size, you can still turn it into a Flash Pool. The SSD space is used to cache the most frequently accessed data. It does not add any user-visible storage capacity to the aggregate.

Does the maximum capacity of the aggregate change when it is converted to a Flash Pool?

Answer: The SSDs do not increase the storage capacity of the aggregate. The SSD space is used to cache the most frequently accessed data. It does not add any user-visible space to the aggregate.

Does Flash Pool make the system faster?

Answer: Flash Pool can make workloads faster by replacing operations that would have gone to HDD with operations to SSD, for both reads and writes. This enables the use of fewer disks, or SATA instead of SAS disks, both of which provide cost efficiency that could not be achieved otherwise. This process requires computational resources and may reduce the maximum size or number of workloads on a system.

How does Flash Pool help write operations?

Answer: Write operations are already accelerated to lower latency than HDD provides through the use of NVRAM technology in the NetApp storage system. NVRAM and its role in the system are unchanged. The SSD in Flash Pool can be thought of as part of the aggregate. NVRAM response times are very low latency. In some high-load or bursty scenarios, the system can enter a situation in which writes occur at the same time as disk rather than being protected in NVRAM, cached in main memory, and responding to the host asynchronously. Flash Pool allows the system to perform random overwrites to a location much faster than HDD, getting out of the synchronous scenario faster or avoiding it altogether.

Are there any special characteristics of the SSD RAID group?

Answer: The SSD raid group can be set up as a RAID 4 or a RAID-DP group, depending on the RAID type of the HDD RAID groups (they must be the same). RAID-level functionality can be considered the

same as any other RAID group. Flash Pool functionality occurs at software levels above RAID; that is, at the WAFL level.

How much usable cache space will be available from the SSD?

Use Table 2 to estimate available cache sizes for Flash Pool created with 100GB SSDs.

Table 2) Flash Pool RAID groups created with 100GB SSDs.

Raid Group Size	Number of Data Disks	Approx. Cache Size
4	2	142GB
6	4	285GB
8	6	427GB
10	8	570GB
12	10	712GB
14	12	855GB
16	14	997GB
18	16	1140GB

Can I mix disk types in a shelf—for example, SATA and SSD in the same shelf?

Answer: You can mix SATA and SSD in the same DS4243 shelf. You cannot mix SSD and SAS drives in the same shelf.

Does Data ONTAP use trim to manage the SSD performance as the SSDs age?

Answer: Yes. Data ONTAP uses the `TRIM` command as well as the native wear-leveling and garbage collection mechanisms on the drive. For information about `TRIM`, see <http://en.wikipedia.org/wiki/TRIM>.

Note: Wikipedia is not associated with NetApp and NetApp is not responsible for content on Wikipedia. This link is provided as a reference and not as an authoritative guide to the `TRIM` command.

Can data be pinned in the SSD portion of the Flash Pool?

Answer: No. There is no pinning of data in the SSD portion of the Flash Pool. Data is maintained in SSD based on use. For reads, this means that recently used data has a higher chance of being retained. For writes, this means that recently written (and overwritten) data has a higher chance of being retained.

What is the granularity of data movement in a Flash Pool?

Answer: Data in the Flash Pool is moved in multiples of 4KB WAFL blocks.

Can a 32-Bit aggregate be converted into a Flash Pool?

Answer: No. The aggregate must be a 64-bit aggregate to enable the Flash Pool functionality.

Can a traditional volume be converted into a Flash Pool?

Answer: No. There is no way to convert a traditional volume into a Flash Pool.

If an aggregate was created with SSDs before Data ONTAP 8.1.1, can that aggregate be converted into a Flash Pool by adding SATA or SAS drives?

Answer: No. The SSDs in that aggregate act as normal data drives and can't be converted to act as cache drives. A new aggregate must be created and the data moved to it. Then the old aggregate can be taken offline and destroyed. Only then can the SSDs be used to convert the new SATA aggregate into a Flash Pool.

Does FlexShare work with Flash Pool?

Answer: Yes, portions do. NetApp recommends not using the priority settings in FlexShare® on a system with Flash Pool. The cache policies can be used to change the behavior of data in the system buffer cache (RAM), but the FlexShare scheduler should not be enabled. To use only the cache option in FlexShare, make the following setting with the priority command: `enabled_components=cache`. For more information about setting cache components only, see TR-3459, FlexShare Design and Implementation Guide.

Does the FlexShare priority mechanism work with Flash Pool to maintain higher-priority volumes in the SSD cache?

Answer: Although you can use FlexShare on a 7-Mode system that has Flash Pool, the FlexShare functionality priority setting does not affect the read or write caching behaviors of insertion or ejection or pinning of data. In addition, the use of volume priorities may artificially limit the performance of the Flash Pool. The `priority hybrid-cache` command is used to change the per-volume insertion policies. NetApp recommends not using FlexShare volume priorities with Flash Pool.

Does the FlexShare Cache=[Reuse|Keep] setting affect Flash Pool?

Answer: Although this setting is used to control the insertion mechanism at a volume level for Flash Cache in Data ONTAP operating in 7-Mode, Flash Pool does not use it. The `priority hybrid-cache` command is used instead.

Is there a Qos capability for data objects in a Flash Pool?

Answer: No. Volume options and a modified form of an LRU algorithm control the insertion policy, at time of insertion only.

Can the amount of cache available to read and write be tuned?

Answer: No. You can tune cache insertion behaviors by using policies at the volume level. Once data is in the cache, the Flash Pool manages data by age and activity, retaining data that is used and active.

Do Flash Pool and SnapLock interoperate?

Answer: No. An aggregate that has been promoted cannot use the NetApp SnapLock® software feature. Likewise, an aggregate with SnapLock data cannot be promoted to a Flash Pool.

Do Flash Pool and NetApp Snapshot interoperate?

Answer: Flash Pool does not cache blocks that belong *only* to Snapshot copies, so it does not improve performance to data that exists only in Snapshot and not in the active file system. All of the standard Snapshot functionality still applies and works in Flash Pool.

Can a Flash Pool's SSDs be removed and the Flash Pool be converted into a regular HDD aggregate?

Answer: No. A Flash Pool cannot be reversed to a normal aggregate. The data must be copied to another location and the Flash Pool aggregate destroyed and rebuilt. The data can then be copied back.

Refer to the [Interoperability Matrix Tool](#) (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

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.

[Go further, faster®](#)



www.netapp.com

© 2012 NetApp, Inc. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, Data ONTAP, FlexShare, RAID-DP, SnapLock, SnapMirror, Snapshot, SnapVault, and WAFL are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. TR-4470-0512