Technical Report

Flash Pool Design and Implementation Guide

Skip Shapiro, NetApp
March 2014 | TR-4070

Abstract

This technical report covers NetApp® Flash Pool™ intelligent caching to provide a firm understanding of how Flash Pool technology works, when and how to use it, and best practices and design considerations for deploying Flash Pool aggregates.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

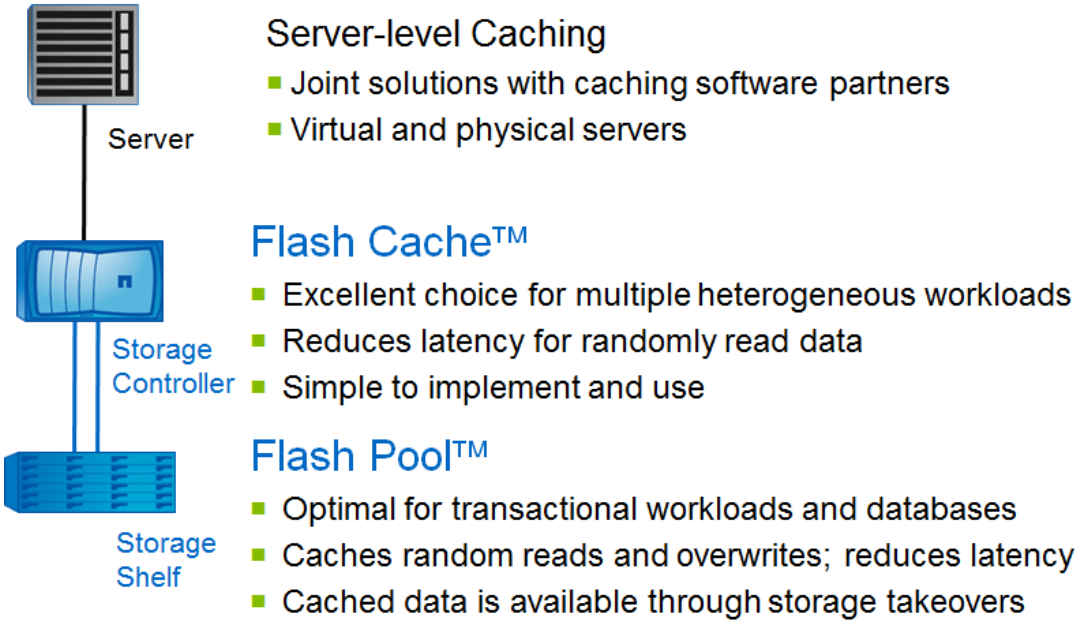
1 Overview

NetApp Flash Pool is an intelligent storage caching product within the NetApp Virtual Storage Tier (VST) product family. A Flash Pool aggregate configures solid-state drives (SSDs) and hard disk drives (HDDs) into a single storage pool (*aggregate*), with the SSDs providing a fast-response-time cache for volumes that are provisioned on the Flash Pool aggregate.

1.1 Virtual Storage Tier

The NetApp VST, as illustrated in Figure 1, offers a portfolio of flash-based products and solutions that provide end-to-end data caching for deployments that use NetApp FAS or V-Series storage systems. The NetApp VST family includes two complementary Data ONTAP® intelligent caching products: NetApp Flash Pool, which uses a hybrid aggregate of SSDs and HDDs, and NetApp Flash Cache™, which uses flash PCIe cards installed in a storage controller. Flash Pool and Flash Cache can also be used with server-based caching software offered by NetApp solution partners. This technical report is focused on Flash Pool.

Figure 1) The NetApp VST product family.



1.2 Flash Pool

A NetApp Flash Pool aggregate configures SSDs and HDDs—either performance disk drives (often referred to as SAS or FC) or capacity disk drives (often called SATA)—into a single aggregate. (An *aggregate* is a NetApp term for a storage pool.) The SSDs are used to cache data for all volumes that are provisioned on the aggregate. Provisioning a volume in a Flash Pool aggregate can provide one or more of the following benefits:

- **Persistent low read latency for large active datasets.** NetApp systems configured with Flash Pool can cache up to 100 times more data than configurations that have no supplemental flash-based cache, and the data can be read 2 to 10 times faster from the cache than from HDDs. In addition, data cached in a Flash Pool aggregate is available through planned and unplanned storage controller takeovers, enabling consistent read performance throughout these events.
- **More HDD operations for other workloads.** Repeat random read and random overwrite operations utilize the SSD cache, enabling HDDs to handle more reads and writes for other workloads, such as sequential reads and writes.

- **Increased system throughput (IOPS).** For a system where throughput is limited due to high HDD utilization, adding Flash Pool cache can increase total IOPS by serving random requests from the SSD cache.
- **HDD reduction.** A storage system that is configured with Flash Pool to support a given set of workloads typically has fewer of the same type of HDD, and often fewer and lower-cost-per-terabyte HDDs, than does a system that is not configured with Flash Pool.

Although configuring a NetApp storage system with Flash Pool can provide significant benefits, there are some things that Flash Pool does not do. For example:

- **Accelerate write operations.** The NetApp Data ONTAP® operating system is already write-optimized through the use of write cache and nonvolatile memory (NVRAM or NVMEM). Flash Pool caching of overwrite data is done primarily to offload the intensive write operations of rapidly changing data from HDDs.
- **Reduce or alleviate high CPU or memory utilization.** Adding a caching technology to a storage system results in an incremental increase in CPU and memory consumption. Consequently, adding Flash Pool to a system that is already near maximum CPU or memory utilization increases the consumption of these resources.
- **Cache sequential (read or write) or large-block (>16KB) random write operations.** HDDs handle sequential read and write operations efficiently. Large-block random write operations are typically organized into more sequential write operations by Data ONTAP before being written to disk. For these reasons and others discussed later in this document, Flash Pool does not cache sequential writes or random overwrites that are larger than 16KB.
- **Increase the maximum throughput capability of a storage system.** Achieving the maximum throughput (IOPS or MB/sec) of a system is a function of the memory and CPU resources of the storage controller. Maximizing throughput also requires a sufficient number of drives (HDDs or SSDs) to handle the workloads that will result in peak system (controller) performance. Caching technologies do not increase the system memory or CPU cycles available in a system. As a result, the maximum throughput values for NetApp storage systems are not higher for systems configured with a caching technology.

2 How Flash Pool Works

Flash Pool is specifically targeted at accelerating repeat random read operations and offloading small-block random overwrite operations (which are a specific class of writes) from HDDs. Although the SSD cache in a Flash Pool aggregate is a single physical resource within the aggregate, the read cache and write cache are separate logical entities from a cache policy and data management standpoint.

2.1 Read Caching

Flash Pool read caching caches random read requests of all sizes. Caching of random reads significantly improves read I/O response times for the volumes provisioned in a Flash Pool aggregate that have read caching enabled. (Note: It is possible to exclude volumes from using read cache; this information is covered later in this report.)

The process of read caching can be broken down into three steps: the initial read request, data insertion into the cache, and subsequent read requests.

Initial Read Request

All new data in a volume provisioned in a Flash Pool aggregate is written to HDDs. When a block of data is read for the first time, the block is read from a HDD into a buffer in system memory and then it is sent from the memory buffer to the requesting client or host. The read data located in memory as a result of this operation is a copy of the data that is stored on the HDD.

Cache Insertion

If a data block that is in memory is not read again for an extended period of time, the block is eventually evicted from the memory. If a data block that is designated for eviction from memory originated from a volume provisioned in a Flash Pool aggregate, and it was randomly read, the data block is inserted into the Flash Pool read cache per the default read cache policy. (Data ONTAP places data blocks that are randomly read in memory buffers designated for that type of read data used to hold the data.)

The cache insertion process for Flash Pool involves adding the data block to a pending consistency point operation that writes the block as part of a RAID stripe. The RAID stripe is written to the SSDs in the Flash Pool aggregate where the source volume is provisioned. The data in the SSD cache is a copy of the data block that is stored on a HDD.

Subsequent Read Requests

Any subsequent read requests for the same data block are served from the Flash Pool SSD cache. The data block is copied from SSD cache into a buffer in system memory and it is forwarded to the requesting client or host. (If a copy of a data block already exists in system memory as a result of a recent read from the Flash Pool read cache, then the data is sent to the client or host directly, and no additional copy from the Flash Pool cache is required.)

When a data block is accessed from the SSD cache, the cache priority of the block is elevated such that it remains in the Flash Pool read cache longer. If a data block is not read from the Flash Pool read cache, it eventually is evicted. When a data block has been evicted from the Flash Pool read cache, the next time a read of that data block is requested, the three-step read caching process starts again with a read from a HDD. (See section 2.3, “Eviction Scanner,” for more information about priority and eviction.)

Sequential Reads

Sequentially read data is not inserted into the Flash Pool read cache, and there is no user-configurable option to enable it. There are two principal reasons for this: First, sequential data is served very efficiently from HDDs with the read-ahead caching algorithms coded in Data ONTAP. This means there is little or no benefit from caching sequentially read data in SSDs. Second, given the limited size of the cache as compared with the capacity of HDDs in the aggregate, caching sequential data can result in cache “blowout.” Cache blowout means that the amount of data inserted into the cache is considerably greater than the cache can effectively store such that active and continuous eviction or destaging of data occurs. The result is that the data that is most valuable to keep in the cache isn’t retained long enough to provide much benefit.

2.2 Write Caching

Flash Pool write caching is targeted at caching overwrites of random data where the operation size is 16KB or smaller. Caching small-block random overwrites offloads write operations, which can consume many HDD I/O cycles, for data that is likely to be invalidated by an overwrite.

The process of overwrite caching can be broken down into three steps: initial random write, overwrite cache insertion, and subsequent reads and overwrites.

Initial Random Write

Data that arrives from a client or host to be stored is first written into a buffer in system memory, and a copy of the data, or the write operation, is also written to nonvolatile memory (NVRAM or NVMEM). Data ONTAP organizes write data while it resides in memory, so that the data is written to storage media in a layout that is as physically sequential as possible. As with random reads, Data ONTAP identifies random writes based on the memory buffers that are used. Only updates of previously written data (overwrites) are considered for caching; consequently, the first random write of a data block to a LUN or volume provisioned in a Flash Pool aggregate is always written to a HDD.

Overwrite Cache Insertion

When a random overwrite of data that was previously written to a Flash Pool aggregate arrives, and the update operation is 16KB or smaller, the overwrite is eligible for insertion into the Flash Pool cache. Whether the overwrite data is actually inserted into the Flash Pool cache depends on how much time has transpired since the previous write and the aging rate of data in the Flash Pool aggregate.

When an overwrite is inserted into the Flash Pool cache, the block containing the previous version of the data is invalidated, and the current valid data resides only on a SSD in the Flash Pool cache. (That is a reason why Flash Pool SSD cache is RAID-protected.) The previous version of data may exist either in the Flash Pool cache or on a HDD.

A workload that has a high mix of recurring random overwrites puts a demanding load on HDDs, and the data is expected to be valid for only a relatively short time before it is updated. Therefore, caching random overwrites on SSDs instead of consuming HDD operations for short-lived data is beneficial. The objective is to store frequently updated data blocks in the Flash Pool cache until the current version of data ages sufficiently that it no longer makes sense to keep it in the cache. When a data block in the cache is colder than the other blocks in the Flash Pool cache, and cache space is needed for fresh data, the aged data block is destaged (copied) to a HDD and then evicted from the Flash Pool cache.

Only small-block random overwrites, and not all small-block random writes, are cached because the goal with the Flash Pool write caching policy is to avoid consuming many HDD operations writing and storing frequently-updated data. Writing data into the Flash Pool cache that is likely to remain valid while it ages to the point at which it is destaged to the HDD to make room for fresher data blocks would only delay writing the data to HDDs; it would not offload or avoid any write operations to the HDDs.

Subsequent Overwrites to and Reads from Write Cache

Additional overwrites of the same data block result in a new data insertion into the Flash Pool cache, and that resets the priority of that data block in the write cache. This is discussed in more detail in section 2.3, “Eviction Scanner.” During the time that a data block resides in the Flash Pool write cache, it can be accessed for read operations, and the response time will be much faster than it would be by reading it from the HDD.

2.3 Eviction Scanner

The eviction scanner is responsible for helping the SSD cache retain the most frequently accessed (hot) data, while making room for new hot data by evicting the least recently used (aged) data blocks. The eviction scanner is activated every 60 seconds to determine whether a scanner process needs to run. Whether the eviction scanner process starts depends on the following criteria:

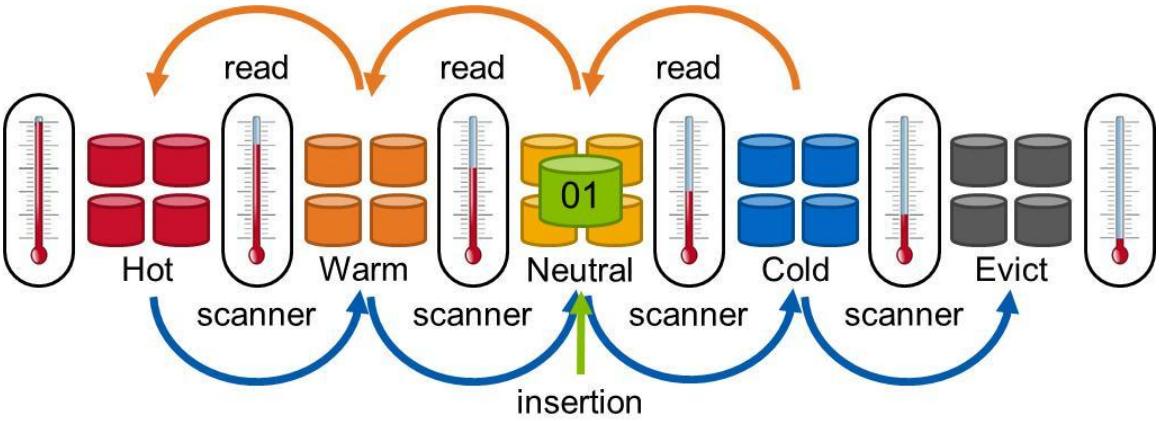
- If the eviction scanner is already running, it does not need to be activated.
- If the SSD cache capacity is used at 75% or higher, the eviction scanner starts and runs to completion.
- If the SSD cache capacity is less than 75% used, and heuristics predict that the cache will reach at least 75% utilization within the next 60 minutes, the eviction scanner starts and runs to completion.
- If none of the preceding conditions is true, then the eviction scanner does not run. The scanner will be activated again in 60 seconds to determine if the scanner process should commence.

There is little or no impact on system performance when the eviction scanner process is running. The read cache and the write cache are managed separately for evictions. Blocks in the cache are assigned a priority level based on a “heat map,” which elevates the priority of frequently accessed data blocks in the read cache and demotes—and eventually evicts—infrequently accessed data blocks in the read cache and data blocks that are not overwritten in the write cache.

Read Cache Evictions

The read cache heat map has five levels of priority ranging from “Hot” to “Evict,” as shown in Figure 2. When a block is inserted into the cache it is assigned to the “Neutral” level. Each time a block is read from the cache, it is promoted to the next higher level in the heat map. Each pass of the eviction scanner demotes a block to the next lower level in the heat map, if the block was not read after the previous eviction scanner pass. When a block reaches the “Evict” level its location in the cache is marked as available for overwriting with new data. A read cache block that is marked for eviction is not destaged (copied) to a HDD because the block is a copy of a block that is stored on a HDD. When a block is read while it is at the “Hot” level, it retains that level because there is no higher priority level in the heat map.

Figure 2) Read cache eviction management.



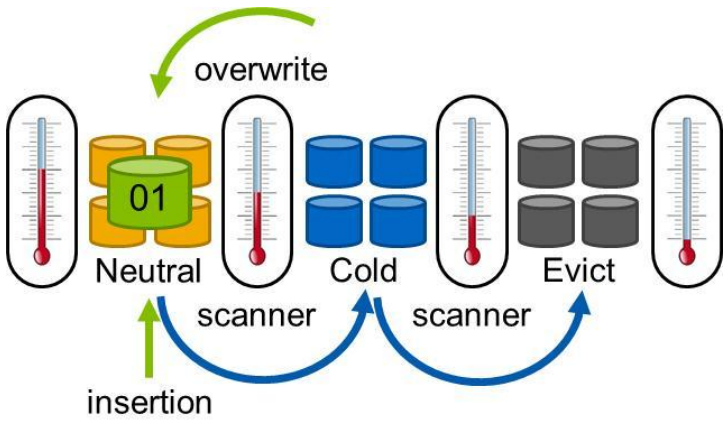
Write Cache Evictions

The write cache has only three levels of priority (unlike the read cache which has five), which range from “Neutral” to “Evict,” as shown in Figure 3. Similar to the read cache, blocks are inserted at the “Neutral” level. The “Hot” and “Warm” levels present in the read cache heat map don’t exist in the write cache heat map. Blocks in write cache can be read—in fact, that is a key benefit of Flash Pool write cache—however, a read does not result in promotion to a higher priority level in the heat map. Blocks are only demoted after a scanner pass.

When a logical block cached in write cache is updated with new data, the updated block is assigned a “Neutral” priority and the physical block that contains the old version of data is marked as invalid (this physical block will be erased in a subsequent operation).

If a block is not overwritten, it will eventually be demoted to the “Evict” level, where it is destaged (written) to a HDD in the aggregate before it is removed from write cache.

Figure 3) Write cache eviction management



2.4 Cache Policies

A key benefit of Flash Pool is that explicit volume-level cache policies can be set. Although Flash Cache with Data ONTAP operating 7-Mode (as a comparison) can prioritize volumes, that is not the same as explicit per-volume cache policies. For example, at a minimum, Flash Cache caches metadata for all volumes provisioned behind a controller, regardless of the volume prioritization. Explicit volume-level cache policies are a valuable feature, given the best-practice recommendations to provision multiple volumes on large-capacity aggregates and to configure a small number of large aggregates per node (controller).

The read cache and write cache have independent Flash Pool cache policies that can be applied either at the aggregate level (for all volumes in the aggregate) or on a volume-by-volume basis. Cache policies must be applied separately for the read cache and write cache. For example, if you want to turn off read and write caching for a specific volume, you have to run two commands, one for the read cache policy change and a second command to change the write cache policy.

The `priority hybrid-cache set <vol/aggr name> <read|write>-cache=<policy>` command is used to modify Flash Pool cache policies. This command must be executed with advanced administrator privileges. In clustered Data ONTAP, this command is accessed through the node shell.

Read Cache Policies

There are four read cache policies:

- **none.** This policy disables the insertion of new data into the cache.
- **random-read (default policy).** This enables caching of all random reads of any block size for a volume, or all volumes, in a Flash Pool aggregate.
- **random-read-write.** This option enables caching of all random reads and random writes of any block size for a volume in a Flash Pool aggregate.

  Note: This is not a write cache policy; rather, copies of random writes are inserted into the read cache, after being written to HDDs, when this policy is enabled. Random writes that are not inserted into the write cache (per the write cache insertion policy) are written to the HDD.

- **meta.** This policy restricts the read cache to caching only metadata blocks.

Although there are several caching policy options for Flash Pool read caching, the two most commonly used policies are `none` and `random-read`. Although it might seem appropriate to use the `meta` cache policy in a high–file count environment, in general the size of the cache is much larger than the amount of metadata present. As a result, setting the cache policy to `meta` will likely not use all of the cache capacity (because user data is excluded).

Best Practice

NetApp recommends using the default Flash Pool read cache policy. The Flash Pool cache dynamically allocates caching capacity based on the I/O demands put upon the volumes in the aggregate. With the exception of excluding one or more volumes from using the cache, the default "`random-read`" cache policy should suffice for the vast majority of workloads.

Write Cache Policies

There are two write cache policies:

- **none.** This policy disables the insertion of new data into the cache.
- **random-write (default policy).** This policy enables caching of random overwrites that are of a block size of 16KB or smaller.

The write cache policies are straightforward: Either the aggregate or a volume is caching random overwrites, or it is not. NetApp recommends using the default Flash Pool read cache policy.

3 Configuration and Administration

This section addresses the requirements for Flash Pool as well as the key “how-to” information.

3.1 Requirements

The requirements for creating a Flash Pool aggregate can be divided into system and aggregate requirements.

System Requirements

The system requirements for Flash Pool are as follows:

- Data ONTAP 8.1.1 operating in 7-Mode or a later release, or clustered Data ONTAP 8.1.1 or a later release, must be in use.
- Both controllers in a HA pair must be running the same version of Data ONTAP.
- Supported platform families as of February 2014 are:
  - FAS2200: All models
  - FAS/V3100: 3160 and 3170
  - FAS/V3200: 3220, 3240, 3250, and 3270
  - FAS/V6000: All models
  - FAS/V6200: All models
  - FAS8000: All models

  Note: Check Hardware Universe for support on platforms introduced after the date of this report.

  Note: FAS/V3070, FAS/V6030, and FAS/V6070 are not supported with Data ONTAP 8.2.

  Note: V-Series systems must use only NetApp SSDs and HDDs within a Flash Pool aggregate.

- Supported disk shelves include: DS4243, DS4246, DS2246, DS4486, and DS14mk4 FC.

  Note: NetApp does not recommend Flash Pool aggregates with DS4486 shelves as a best practice. DS4486 shelves are intended for secondary storage use cases that typically don’t need or benefit from storage caching.

  Note: Flash Pool with DS14mk2 FC shelves is not supported with NetApp MetroCluster™ systems.

- Supported HDD disk types include: performance (10,000 or 15,000 rpm) or capacity (7200 rpm).

  Note: Flash Pool is not supported on systems that use NetApp Storage Encryption (NSE).

- Supported SSDs include: 1.6TB, 800GB, 400GB, 200GB, and 100GB models (see the notes following Table 3 for supported SSD part numbers).

  Note: Check Hardware Universe for support of shelves, HDDs or SSD that introduced after the date of this report.

Aggregate Requirements

The aggregate requirements for Flash Pool are as follows:

- The aggregate must be a 64-bit aggregate (Flash Pool is not supported for traditional volumes or 32-bit aggregates).

  Note: In general, with Data ONTAP 8.1 and later releases, a 32-bit aggregate can be converted into a 64-bit aggregate, and the 64-bit aggregate can then be converted into a Flash Pool aggregate. However, there are situations in which a 64-bit aggregate that was converted from a 32-bit aggregate cannot become a Flash Pool aggregate. Refer to the “Storage

Management Guide" for the version of Data ONTAP in use to confirm whether an aggregate can be converted into a Flash Pool aggregate.

- The aggregate must be a HDD aggregate (it cannot be an SSD aggregate or an aggregate that uses encrypting HDDs).
- The aggregate cannot be in a `FAILED`, `LIMBO`, offline, or foreign state.
- Supported RAID configurations are: RAID 4, NetApp RAID-DP® technology, and NetApp SyncMirror® software (either local SyncMirror or with MetroCluster).

3.2 Creating and Modifying a Flash Pool Aggregate

The three common administrative actions to understand when using Flash Pool are: creating a Flash Pool aggregate, expanding a Flash Pool aggregate, and changing caching policies.

RAID Group Configuration with Flash Pool

As explained earlier in this report, a Flash Pool aggregate consists of separate RAID groups of SSDs and a single type of HDD. The RAID group policy includes the RAID type (for example, RAID-DP or RAID 4) and the RAID group size.

Starting with the Data ONTAP 8.2 software release, the RAID policies for the SSD RAID group (or groups) are independent of the policies for the HDD RAID groups within a Flash Pool aggregate. For example, a SSD RAID group in a Flash Pool aggregate can be configured with RAID 4 and a group size of 8, and the HDD RAID groups in the same Flash Pool aggregate can use RAID-DP with a group size of 16.

However, for releases in the Data ONTAP 8.1 family beginning with Data ONTAP 8.1.1, the RAID policies within a Flash Pool aggregate that apply to SSD RAID groups are identical to the policies that govern the HDD RAID groups. That means if a Flash Pool aggregate uses capacity (that is, SATA) disk drives in RAID-DP groups of size 16, then the SSD RAID group must also use RAID-DP and have up to 16 drives.

Note: When creating a Flash Pool aggregate on a system that is running a Data ONTAP 8.1.x release, it is possible to add an SSD RAID group that has more drives than the HDD RAID group size. This requires changing the RAID group size for the Flash Pool aggregate before adding the SSD RAID group. If an HDD RAID group is subsequently added to the Flash Pool aggregate, it should be done manually to maintain the same number of disk drives as the existing HDD RAID groups.

Note: On a system that is running a Data ONTAP 8.1.x release, a SSD RAID group in a Flash Pool aggregate along with RAID groups of capacity (often referred to as *SATA*) HDDs cannot exceed the maximum allowable RAID group size for capacity HDDs. When using RAID-DP, the maximum allowable RAID group size is 20 drives.

In addition, the following attributes apply to SSD RAID groups in a Flash Pool aggregate:

- The capacity of the SSD data drives is not part of the storage capacity of the Flash Pool aggregate. Only the HDD data drive capacities count against the maximum capacity of the Flash Pool aggregate.
- The SSDs are not subject to aggregate space reservations such as aggregate NetApp Snapshot™ reserve.
- The SSDs are not subject to volume space reservations such as the volume Snapshot reserve.
- Although blocks from volumes in the aggregate are kept in the write cache, there is no specific cache capacity allocated to individual volumes. The SSD cache capacity is a pool of capacity that is available to any volume in the aggregate.
- SSDs and HDDs cannot be mixed in the same RAID group.

Creating a Flash Pool Aggregate

A Flash Pool aggregate can be created non-disruptively, meaning while the system is operating and serving data. The process of creating a Flash Pool aggregate has three steps:

1. Create the 64-bit HDD aggregate (unless it already exists).

   Note: When creating an aggregate of multiple HDD RAID groups, NetApp best practice is to size each RAID group with the same number of drives or with no more than 1 drive difference (for example, one RAID group of 16 HDDs and a second one of 15 HDDs is acceptable).

   Note: If an existing aggregate is 32-bit, it must be converted to a 64-bit aggregate before it is eligible to become a Flash Pool aggregate. As noted in section 3.1, there are situations in which a converted 64-bit aggregate is not eligible to become a Flash Pool aggregate.

2. Set the `hybrid_enabled` option to `on` for the aggregate:

   a. For Data ONTAP operating in 7-Mode, the command is:

   ```
   aggr options <aggr_name> hybrid_enabled on
   ```

   b. For clustered Data ONTAP, the command is:

   ```
   storage aggregate modify –aggregate <aggr_name> -hybrid-enabled true
   ```

3. Add SSDs into a new RAID group for the aggregate; this creates the SSD cache.

   Note: A RAID group cannot be removed from an aggregate after the aggregate has been created. Reverting a Flash Pool aggregate back to a standard HDD-only aggregate requires migrating the volumes to an HDD-only aggregate. After all volumes have been moved from a Flash Pool aggregate, the aggregate can be destroyed, and then the SSDs and HDDs are returned to the spares pool, which makes them available for use in other aggregates or Flash Pool aggregates.

A Flash Pool aggregate that has a SSD RAID group containing one data drive is supported; however, with such a configuration, the SSD cache can become a bottleneck for some system deployments. Therefore, NetApp recommends configuring Flash Pool aggregates with a minimum number of data SSDs, as shown in Table 1.

Table 1) Minimum recommended number of data SSDs per Flash Pool aggregate.

System Family	100GB SSD	200GB SSD	400GB SSD	800GB SSD	1.6TB SSD
FAS2200 series	1	1	1	Not supported	Not supported
FAS/V3100 series	3	2	2	Not supported	Not supported
FAS/V3200 series[1]	3	2	2	2[2]	2[3]
FAS8020 / FAS8040	3	2	2	2	2
FAS/V6000 series	9	5	5	Not supported	Not supported
FAS/V6200 series	9	5	5	5	5
FAS8060	9	5	5	5	5

Note: (1) Flash Pool is not supported with FAS/V3210; Flash Pool is supported all other models.

Note: (2) 800GB SSDs are supported with FAS/V3220, FAS/V3250, and FAS/V3270 starting with Data ONTAP 8.2.

Note: (3) 1.6TB SSDs are supported with FAS/V3250 only.

Note: At least one hot spare SSD per node (controller) is required when using RAID 4, and one hot spare is strongly recommended when using RAID-DP.

Expanding a Flash Pool Aggregate

When expanding the capacity of a Flash Pool aggregate by adding HDDs, NetApp recommends adding full RAID groups of the same size as the existing HDD RAID groups in the Flash Pool aggregate.

When increasing the size of the SSD cache in a Flash Pool aggregate by adding SSDs to an existing SSD RAID group, NetApp recommends adding at least the number of SSDs shown in Table 2.

Table 2) Minimum recommended number of SSDs to add when expanding an existing SSD RAID group.

System Family	100GB SSD	200GB SSD	400GB SSD	800GB SSD	1.6TB SSD
FAS2200 series	1	1	1	Not supported	Not supported
FAS/V3100 series[1]	3	2	2	Not supported	Not supported
FAS/V3200 series[2]	3	2	2	2[3]	2[4]
FAS8020 / FAS8040	3	2	2	2	2
FAS/V6000 series	6	3	3	Not supported	Not supported
FAS/V6200 series	6	3	3	3	3[5]
FAS8060	6	3	3	3	3

Note: (1) Flash Pool is supported with the FAS/V3160 and FAS/V3170.

Note: (2) Flash Pool is not supported with FAS/V3210; Flash Pool is supported with other models.

Note: (3) 800GB SSDs are supported for Flash Pool use with FAS/V3220, FAS/V3250, and FAS/V3270 starting with Data ONTAP 8.2.

Note: (4) 1.6TB SSDs are supported with FAS/V3250 only for Flash Pool use.

Note: (5) 1.6TB SSDs are supported on all models except the FAS/V6210 for Flash Pool use

Changing Cache Policies

The `priority` command is used to modify Flash Pool cache policies for an individual volume. In clustered Data ONTAP, the `priority` command is accessed through the node shell.

The syntax of the `priority` command is as follows:

- `priority hybrid-cache set <vol_name> <read|write>-cache=<policy>`
  - Where `<vol_name>` is the name of a volume within a Flash Pool aggregate.
  - Where `<read|write>` is used to specify whether the cache policy change applies to the read cache or the write cache (because they are managed separately).
  - Where `<policy>` is any valid policy (covered in section 2.4, “Cache Policies”).

Note: The `priority` command can be used in diagnostic mode to change a cache policy for all volumes in a Flash Pool aggregate with a single command. Diagnostic mode should be used only under the guidance of NetApp technical field or support personnel.

3.3 Maximum Cache Capacities

The maximum size of the Flash Pool cache per system is a function of the controller model and the Data ONTAP release in use. In addition, when both Flash Pool and Flash Cache are used on the same system, the maximum cache size for each caching solution applies, as does the total maximum cache value for the combined use. Hardware Universe lists the supported maximum cache sizes for Flash Cache, Flash Pool, and their combined use by platform (controller) model and Data ONTAP release.

Here is an example of how to apply the maximum cache sizes using the values for a FAS8020 running clustered Data ONTAP 8.2.1 from *Hardware Universe*:

Table 3) Maximum cache sizes for a FAS8020 running clustered Data ONTAP 8.2.1.

	Maximum Cache per Node	Maximum Cache per HA Pair
Flash Cache only	1.5 TiB	3 TiB
Flash Pool only	6 TiB	6 TiB
Combined Flash Cache and Flash Pool	6 TiB	6 TiB

If each node of the FAS8020 is configured with 1.5 TiB of Flash Cache—making a total of 3 TiB of Flash Cache for the HA pair—then up to 3 TiB of Flash Pool cache can be added to the HA pair. The Flash Pool cache can be placed evenly on each node, configured entirely on one node, or distributed in some intermediate configuration.

Generalizing, the guidelines are as follows:

- On a system that is configured with only Flash Cache, do not exceed the Flash Cache per-node maximum cache sizes.
- On a system that is configured with only Flash Pool:
  - If Data ONTAP 8.1.1 or a later 8.1.x release is being used, do not exceed the Flash Pool per-node maximum cache sizes.
  - If Data ONTAP 8.2 or a later 8.2.x release is being used, do not exceed the Flash Pool per-HA-pair maximum cache sizes. There are no separate per-node maximum cache sizes.
- On a system that is configured with both Flash Cache and Flash Pool:
  - If Data ONTAP 8.1.1 or a later 8.1.x release is being used, do not exceed the individual Flash Cache and Flash Pool per-node maximum cache sizes, and do not exceed the combined per-node or per-HA-pair maximum cache sizes.
  - If Data ONTAP 8.2 or a later 8.2.x release is being used, do not exceed the individual Flash Cache per-node maximum cache size, and do not exceed the combined per-HA-pair maximum cache sizes.

The maximum number of SSD data drives supported in one or more Flash Pool aggregates on a system is based on the cache maximum of each platform model, which is found in Hardware Universe. Table 4 shows the maximum number of SSD data drives by drive model and Flash Pool cache sizes, using cache sizes that correspond to the maximum sizes for the range of platforms and Data ONTAP releases in Hardware Universe.

Note: Table 4 reflects the maximum number of SSD data drives per Flash Pool or system. SSDs used for RAID parity and hot spares are in addition to the values in the table, except for when 100GB SSDs are used on a system that has a maximum of 24TiB of Flash Pool cache. For this exception, 240 SSDs total, including parity and hot spares, is the maximum.

Table 4) Maximum number of SSD data drives per maximum Flash Pool cache sizes.

Flash Pool Max Cache Size	100GB Data SSDs [1]	200GB Data SSDs [2]	400GB Data SSDs [3]	800GB Data SSDs [4]	1.6TB Data SSDs [5]
300 GiB	3	1	Not supported	Not supported	Not supported
400 GiB	4	2	1	Not supported	Not supported
0.5 TiB	5	2	1	Not supported	Not supported
0.6 TiB	6	3	1	Not supported	Not supported
800 GiB	8	4	2	Not supported	Not supported
1.0 TiB	11	5	2	Not supported	Not supported
1.2 TiB	13	6	3	Not supported	Not supported
1.6 TiB	17	8	4	Not supported	Not supported
2.0 TiB	22	11	5	Not supported	Not supported
4.0 TiB	44	22	10	5	2
6.0 TiB	66	33	16	8	4
12.0 TiB	132	66	32	16	8
18.0 TiB	198	99	49	24	12
24.0 TiB	240 [6]	132	65	32	16

Note: (1) 100GB SSD part number X441A-R5 with a right-sized capacity 95,146 MiB.

Note: (2) 200GB SSD part numbers X446A-R6, X446B-R6, and X448A-R6 with right-sized capacity 190,532 MiB.

Note: (3) 400GB SSD part numbers X438A-R6 and X575-R6 with right-sized capacity 381,304 MiB.

Note: (4) 800GB SSD part numbers X447A-R6 and X449A-R6 with right-sized capacity 762,847 MiB.

Note: (5) 1.6TB SSD part numbers X439A-R6 and X576A-R6 with right-sized capacity 1,525,935 MiB.

Note: (6) The maximum number of SSDs supported on FAS/V 3xx0, 6xx0, and 8000 systems is 240.

3.4 AutoSupport Reporting

Flash Pool data is included as part of the information provided in basic NetApp AutoSupport™ (ASUP™) reporting. In addition to information specific to Flash Pool, SSD write wear information is also available for any SSD attached to the system (regardless of whether it is a spare, in a Flash Pool aggregate, or in a pure SSD aggregate).

4 Best Practice Recommendations and Interoperability

Flash Pool works with most features of Data ONTAP. This section covers guidelines for using Flash Pool, along with some of the most widely used features of Data ONTAP, as well as important deployment considerations and best practices.

4.1 Deployment Considerations

Several things should be considered when deploying a Flash Pool aggregate.

SSD RAID Protection and Spares

Although the SSDs in a Flash Pool aggregate are used primarily to cache a copy of blocks stored on HDDs, data blocks in the Flash Pool write cache hold the only copy of that data. This is a principal reason why the SSDs in a Flash Pool aggregate are RAID protected. Let's examine some considerations for SSD RAID group and hot spare policies.

As explained in section 3.2, "Creating and Modifying a Flash Pool Aggregate," when using a Data ONTAP release earlier than Data ONTAP 8.2, there is a single RAID group policy for a Flash Pool aggregate that applies to both the SSD RAID group and the HDD RAID groups. The policy is based on the type of HDD (for example, SAS or SATA) used in the Flash Pool aggregate. NetApp recommends using RAID-DP (RAID double-parity) protection for all HDD RAID groups, and thus the SSD RAID group is also RAID-DP protected. (RAID-DP protects data when two drives in the same RAID group are in a failed state or have uncorrectable errors simultaneously.)

When using Data ONTAP 8.2 or a subsequent release, the RAID policies for the SSD RAID group and HDD RAID groups in a Flash Pool aggregate are independent. That means an SSD RAID group could be RAID 4 protected, while the HDD RAID groups in the same Flash Pool aggregate use RAID-DP protection. Nevertheless, the added protection of RAID-DP makes it a best practice to use RAID-DP for SSD RAID groups as well. An uncorrectable error in an SSD RAID group that is configured with RAID 4 and has experienced the failure of one SSD results in the entire Flash Pool aggregate being taken offline. It could also cause a loss of data that is cached in the write cache. Therefore, NetApp recommends using RAID-DP protection for SSD RAID groups and HDD RAID groups.

NetApp also recommends maintaining at least one spare SSD for each node (controller) that has a Flash Pool aggregate configured on it. However, on a FAS2200 series system that is running a Data ONTAP 8.1.x release, it is acceptable to configure the SSD RAID group in the Flash Pool aggregate with RAID-DP and to operate without a hot spare SSD.

Note: The RAID option `raid.min_spare_count` does not apply to Flash Pool SSD RAID groups.

On a FAS2200 series system that is running Data ONTAP 8.2, the recommendation is to configure the SSD RAID group with RAID 4 protection and one hot spare SSD. With this configuration, data is protected, and an immediate and fast RAID group rebuild occurs if a SSD fails. If RAID 4 protection is chosen for an SSD RAID group in a Flash Pool aggregate on any other FAS or V-Series system, at least one hot spare SSD should be maintained for each node that has a Flash Pool aggregate configured on it.

Number of Volumes per Flash Pool Aggregate

To maximize the performance benefit of Flash Pool, NetApp recommends provisioning three or more volumes in a Flash Pool aggregate. This enables a multicore controller to process I/O to and from the Flash Pool aggregate more effectively.

Number of Flash Pool Aggregates per System

There is no specified maximum number of Flash Pool aggregates per system, although technically the limit is the maximum number of aggregates allowed on the system. However, given the finite maximum

Flash Pool cache sizes (as indicated in Table 4), there are practical limits for the number of Flash Pool aggregates that can be deployed while providing optimal performance benefit.

For storage system deployments in which the random read-write workload of individual volumes might vary considerably or is not well known, configuring fewer Flash Pool aggregates that have larger cache sizes typically results in more effective utilization of the cache and better overall performance.

In contrast, if there are many volumes that each have a consistently high random read-write workload and a working set (active portion of the dataset) that is well understood and large enough to consume 0.5 TiB of cache or more, then dividing the workloads across multiple Flash Pool aggregates is advised.

Finally, when considering the number of Flash Pool aggregates to configure and how much cache each one should have, follow the recommended minimum number of SSD data drives per Flash Pool aggregate, as shown in Table 1, and the maximum number of SSD data drives per Flash Pool aggregate, as shown in Table 4.

Snapshot Copies

When a volume Snapshot copy is taken, the virtual block numbers for the blocks included in the Snapshot copy are locked. When a block stored in Flash Pool write cache that is also part of a Snapshot copy is evicted and destaged (written) to a HDD, the virtual block number remains the same, and only the physical block number is updated.

An aggregate Snapshot copy locks the physical block numbers for the blocks included in the Snapshot copy. For a block that is stored in Flash Pool write cache, an aggregate Snapshot copy locks the physical block number in the SSD cache until the Snapshot copy is deleted. Therefore, NetApp recommends disabling aggregate Snapshot copies for Flash Pool.

Note: MetroCluster systems require the use of aggregate Snapshot copies. Flash Pool aggregates are recommended with MetroCluster systems because the aggregate Snapshot copies are typically short-lived, and thus Flash Pool cache physical block numbers are locked in a Snapshot copy only for a short period of time.

4.2 Using Flash Pool and Flash Cache

Flash Cache and Flash Pool can be used together on one node (controller), within a HA pair, and across multiple nodes in a clustered Data ONTAP cluster. Data from volumes that are provisioned in a Flash Pool aggregate or an all-SSD aggregate is automatically excluded from being inserted into Flash Cache cache. Data from volumes provisioned in a Flash Pool aggregate uses the Flash Pool cache, and data from volumes provisioned on a SSD aggregate does not benefit from using Flash Cache.

Note: Disabling Flash Pool or setting the volume or Flash Pool cache policies to `none` does not enable data from volumes provisioned in the Flash Pool aggregate to be cached in Flash Cache. Data is excluded from Flash Cache based on the type of aggregate on which a volume is provisioned.

The key attributes of Flash Cache are:

- Randomly read data is automatically cached for all volumes on a node that are provisioned on HDD-only aggregates.
- It is simple to deploy and use.
- Cache capacity does not affect the maximum number of drives that can be configured on a system.

The key attributes of Flash Pool are:

- Randomly read and randomly overwritten data for all volumes provisioned in Flash Pool aggregate can be automatically cached.
- Cached data is accessible during planned and unplanned storage controller takeover and giveback events, and remains intact and valid after a controller reboots.

- Explicit cache policies can be applied to all volumes in a Flash Pool aggregate or separately for each volume.

Generally, Flash Cache and Flash Pool provide similar storage efficiency and performance benefits for workloads that consist predominantly of random read operations. This means other factors are usually more important in deciding which of the two caching solutions to use. There are deployment considerations and environments in which one caching solution is preferred over the other, as well as a few situations in which only one solution fits.

Use cases where Flash Pool is the only supported caching solution are with FAS2220 and FAS2240 systems. Examples of where Flash Cache is the only caching solution available are with FAS/V3210 and FAS/V3140 systems; Flash Cache is supported with the Data ONTAP 8.1.1 and later 8.1.x releases, however Flash Pool is not supported with these systems.

For other use cases, Table 5 provides general guidelines to help determine when to use Flash Cache and when to use Flash Pool.

Table 5) General guidelines for Flash Cache and Flash Pool use.

Flash Cache is the preferred solution when one or more of the following conditions apply:	Flash Pool is the preferred solution when one or more of the following conditions apply:
• A system has applications or workloads that have a high mix of random reads, and cached data access during controller takeovers is not required. • FAS FlexArray and V-Series systems in which extended caching of read data from LUNs provisioned on third-party storage is needed. • Caching sequential data that will be read repeatedly is needed. • NetApp Storage Encryption (NSE) systems in which extended caching of read data is needed.	• An application or workload has a high mix of random reads, and it requires consistent performance through controller takeover events. • An application or workload has a high mix of random write updates that are 16KB or smaller in size. • Dedicated cache for some volumes is desired. • Cache sizes that are larger than Flash Cache allows are needed.

Note: NetApp recommends using only one caching solution—either Flash Cache or Flash Pool, but not both together—on systems for which the combined maximum Flash Cache and Flash Pool cache size per node (controller) is 2TiB or smaller. (Refer to Hardware Universe to see which system and Data ONTAP release combinations apply to this recommendation.)

4.3 Storage Efficiency

Flash Pool works with Data ONTAP deduplication and NetApp FlexClone® technology, and storage efficiency is preserved in the Flash Pool SSD cache. Only one physical 4KB block on a Flash Pool SSD is needed to cache many duplicate or shared (cloned) logical references to identical data. Integration with deduplication and FlexClone enables Flash Pool to cache more unique logical data, thereby accelerating more read accesses and offloading more HDD operations.

Volumes on which compression is enabled can be provisioned in a Flash Pool aggregate. However, compressed blocks are not cached in Flash Pool cache. A volume that has compression enabled can contain both compressed and uncompressed blocks; only the uncompressed blocks can be inserted into Flash Pool cache.

Compressed blocks that are read from HDDs are uncompressed and then placed into memory read buffers. Compressed and uncompressed blocks have different data formats. Section 2.1, “Read Caching,” explains that blocks that are marked for eviction from read buffer memory are inserted into the Flash Pool read cache according to the read caching policy that is in effect. Data in Flash Pool read cache must be in

the same format as the data stored on the HDDs, and this is not possible when the data on the HDDs is compressed.

If a block that is cached in the Flash Pool read cache is subsequently compressed on a HDD, the cached block is invalidated and eventually evicted from the Flash Pool read cache.

4.4 Data Protection

Flash Pool caching works with the built-in data protection capabilities of Data ONTAP. NetApp SnapMirror® replication technology can be used to replicate data from or to a volume that is provisioned in a Flash Pool aggregate. The same is true when using NetApp SnapVault® software for disk-to-disk backup. However, Flash Pool caches blocks only from volumes that have read-write access; blocks from read-only volumes—such as SnapMirror or SnapVault destination volumes—are not cached. Data stored in a read-only volume can be cached by creating a read-write clone of the volume using FlexClone software, and then using the clone for read or read-write access.

4.5 High-File-Count Environments

High-file-count (HFC) environments are characterized by a large amount of metadata as compared with user or application data. A common question with respect to HFC environments and Flash Pool (and for Flash Cache as well) is whether caching only metadata is more effective than caching both metadata and user data (which is the default reach caching policy).

Data ONTAP storage systems that are used to handle HFC workloads benefit from using Flash Pool (or Flash Cache) to cache the large amount of metadata that is read repeatedly. NetApp recommends using the default `random-read` read caching policy for HFC datasets. This cache policy enables both active metadata and active user data to be cached as needed and typically provides the best overall performance.

5 Performance Expectations and Flash Pool Statistics

5.1 Performance Improvement When Using Flash Pool

It is important to understand the performance improvements that Flash Pool can provide as a function of storage system workloads and configuration. The potential benefits from provisioning volumes in a Flash Pool aggregate, as described in section 1.2 of this report, are repeated below and then explained in more detail:

- Persistent fast read response time for large active datasets
- More HDD operations for other workloads
- Increased system throughput (IOPS)
- HDD reduction

Random read requests that are serviced with data cached in a Flash Pool cache are typically delivered with 3- to 4-millisecond latency; this is roughly 2 to 6 times faster than latencies for random reads from HDDs. Faster servicing of data can increase client or host throughput because the client or host experiences less idle time waiting for data, and it can also result in better application response times. These performance benefits can be sustained through planned and unplanned storage controller takeover events.

In addition to improving application response times and client or host throughput, random read and write requests that are handled by a Flash Pool cache offload HDDs from servicing those requests. That allows the HDDs to service other I/Os—for example, sequential reads and writes—which in turn might enable the storage system to handle more workload (that is, deliver higher throughput). An increase in storage system throughput depends on whether system performance is gated by HDD throughput. If the

aggregated throughput of the HDDs is limiting system throughput, offloading requests to a Flash Pool SSD cache from HDDs should increase overall storage system throughput.

The benefit of offloading I/O requests from HDDs to a Flash Pool SSD cache can be realized in a different way. If a storage system is configured to handle a known set of workloads, including some that consist of random reads and writes, a system configured with a Flash Pool SSD cache typically requires fewer HDDs than does a configuration that has neither Flash Pool nor Flash Cache—if performance rather than storage capacity is determining the total number of HDDs on the system. When this is the case, the result is often that the system configured with Flash Pool has a lower cost than the alternative system that is configured only with HDDs.

5.2 Flash Pool Statistics and Performance Monitoring

As stated earlier, Flash Pool caches random read requests, and repeat overwrite requests that are 16KB or smaller in size. Consequently, a dataset that has a workload with a high mix of either or both of these requests benefits most from being provisioned in a Flash Pool aggregate.

The performance benefit that an existing Flash Pool aggregate provides can be monitored by using the `stats show` command with the preset `hybrid_aggr`. The full command syntax is `stats show –p hybrid_aggr`, and using it requires the advanced-privilege administration level.

Note: The `hybrid_aggr` preset is available for use starting with the Data ONTAP 8.1.2 and 8.2 releases.

Note: For systems running clustered Data ONTAP, the `statistics` command is executed from the node shell (`run –node –node01`, for example).

Figure 4 shows an excerpt of a typical output from the `stats show –p hybrid_aggr` command. Descriptions of the 14 columns of data displayed in the output are listed in Table 6.

Figure 4) Flash Pool statistics output.

Instance	ssd blks used	blks rd cached	blks wrt cached	read ops replaced /s	read ops replaced %	write blks replaced /s	write blks replaced %	rd cache evict /s	wr cache destage /s	rd cache ins rate /s	wr cache ins rate /s	read hit latency	read miss latency
flash_aggr	52326120	25374159	17624092	9438	88	0	0	5582	5915	2844	3320	5.61	14.68
flash_aggr	52331900	25368605	17616100	8836	88	0	0	4710	6197	3181	3392	5.85	14.84
flash_aggr	52334712	25365170	17608177	8277	88	0	0	3159	5289	3132	3080	6.42	14.05
flash_aggr	52339765	25364531	17608212	9478	89	0	0	4192	6015	3748	3526	5.56	14.36
flash_aggr	52347675	25357944	17605207	10864	89	0	0	4931	6175	1467	1513	4.45	13.14
flash_aggr	52347417	25352786	17595598	10211	89	2187	16	258	31	2762	2219	5.22	14.98
flash_aggr	52350696	25354544	17594364	12657	90	4967	43	0	0	2758	4967	3.47	13.56
flash_aggr	52357194	25357295	17599044	12619	89	6657	51	0	0	2905	6657	3.81	14.28
flash_aggr	52366556	25354910	17600111	11514	89	2134	30	0	0	3791	2134	4.13	13.85
flash_aggr	52370836	25355974	17597018	11704	90	4409	44	0	0	4731	4409	3.85	14.04

Columns 2 through 4 indicate how many data blocks have been cached and whether they were cached by the read or the write cache policy. Columns 5 through 8 show the number of operations or blocks serviced by the SSD cache instead of HDDs and the resulting replacement or hit rates. Higher replacement rates mean the cache is more effective at offloading I/O from HDDs and it is providing greater performance benefit.

Columns 9 through 12 provide the rate at which blocks are inserted or removed from the cache, and they indicate how actively data is flowing into and out of the cache. High eviction or destaging rates relative to cache hit rates or insertion rates suggests that increasing the size of the cache could provide more benefit.

Finally, column 13 shows the average latency for read requests served from the SSD cache, and column 14 shows the average latency for read requests served from HDDs, which is the same as a read miss from the cache. In this sample output, cache hit latencies are 2 to 4 times faster than cache miss latencies are.

When assessing Flash Pool performance, it is important to confirm that the cache is warmed up and that steady-state operation has been achieved. When a Flash Pool aggregate is initially created, or data SSDs have been added to an existing cache, it takes time to warm up the cache. With a new Flash Pool aggregate, most of the initial operations are serviced by HDDs. As read and write data is inserted into the cache, the number of operations serviced from the cache should increase until steady-state behavior is observed in statistics such as SSD blocks used, rates for read operations and write block replacements, and read hit latency. Depending on the workload level and the size of the Flash Pool cache that has been deployed, it might take many hours to reach a steady-state condition.

The default time interval for each entry in the statistics output is 5 seconds. The time interval can be changed by using the `-i` switch and specifying the interval in seconds. For example, `stats show -p hybrid_aggr -i 1` would change the output interval to every 1 second.

Note: Although the interval of the command output might be more than 1 second (5 seconds by default), it is important to note that the unit of measure for the counters is per second. For example, using the default 5-second interval, the results shown would be the per-second average over the previous 5 seconds (interval of collection).

Table 6) Definition of Flash Pool statistics from the stats show –p hybrid_aggr command.

Statistic	Description
`Instance`	The Flash Pool aggregate that this row of output describes.
`ssd blks used`	The amount of data currently in the SSD cache. This includes read cache data, write cache data, and Flash Pool metadata.
`blks rd cached`	The amount of data, in 4KB blocks, currently in the read cache.
`blks wrt cached`	The amount of data, in 4KB blocks, currently in the write cache.
`read ops replaced rate /s`	The number of disk read operations replaced per second. This is actual disk I/Os, not 4KB blocks. A read operation can contain more than one 4KB block.
`read ops replaced rate %`	The percentage of disk read operations replaced, on a per-second interval, averaged from one line of output to the next.
`write blks replaced rate /s`	The number of 4KB blocks written to SSDs instead of to HDDs.
`write blks replaced rate %`	The percentage of write blocks written to SSDs instead of to HDDs, on a per-second interval, averaged from one line of output to the next.
`rd cache evict /s`	The rate that data was evicted from read cache, in 4KB blocks per second.
`wr cache destage /s`	The rate that data was destaged from write cache to HDDs, in 4KB blocks per second.
`rd cache ins rate /s`	The rate that data was inserted into read cache, in 4KB blocks per second.
`wr cache ins rate /s`	The rate that data was inserted into write cache, in 4KB blocks per second.
`read hit latency`	The latency, in milliseconds, of read operations that are successfully served from the Flash Pool SSD cache.
`read miss latency`	The latency, in milliseconds, of read operations that are not served from Flash Pool SSD cache and must be read from HDDs instead.

6 Workload Analysis and Cache Sizing

There are several methods of analyzing the workloads on a FAS or V-Series system to determine whether introducing Flash Pool cache, or expanding the size of an existing Flash Pool cache, might be beneficial. The following bullets describe four methods, and two of them are explained in more detail in this section of the report.

- For a system that is using Data ONTAP 8.2.1 or a later release, the Automated Workload Analyzer (AWA) feature of the software can be used. AWA works on systems that have Flash Cache or Flash Pool configured, as well as on systems that use neither caching product. AWA is explained further in section 6.1.
- For a system that is using a release earlier than Data ONTAP 8.2.1 and that has one or more Flash Pool aggregates provisioned, the Flash Pool statistics described in section 5.2 can be used to estimate the optimal cache size for each Flash Pool aggregate.
- For a system that is using a release earlier than Data ONTAP 8.2.1 and that has neither Flash Pool nor Flash Cache provisioned, Predictive Cache Statistics (PCS) can be used to assess the benefit of adding Flash Pool and to estimate the cache size. Using PCS is covered in Section 6.2.
- For a system that is using a release earlier than Data ONTAP 8.2.1 and that has Flash Cache installed, NetApp recommends making the cache size of a Flash Pool aggregate that will be added to the system at least equal to the total cache size of the existing Flash Cache cards on the node.

6.1 Sizing Flash Pool Cache with Automated Workload Analysis

The Automated Workload Analyzer (AWA) feature, which is available starting with Data ONTAP 8.2.1, analyzes the read-write workload mix and the percentage of reads and writes that are cacheable, on a per-aggregate basis. This makes AWA an excellent way to analyze workloads for Flash Pool consideration.

AWA analyzes the aggregated workload of all volumes and LUNs in an aggregate, recommends a Flash Pool cache size for the aggregate, and estimates the cache hit rates for the recommend cache size as well as smaller increments of cache. AWA is available with both clustered Data ONTAP and Data ONTAP operating in 7-Mode. It runs from the node shell command line and requires advanced privilege administration access to the system.

AWA consists of three commands:

- `wafl awa start <aggrname>`
- `wafl awa stop <aggrname>`
- `wafl awa print <aggrname>`

The `wafl awa start` command is used to begin the collection of workload statistics.

The `wafl awa stop` command is used to end the collection of workload statistics and print a summary report.

The `wafl awa print` command is used to print a summary report while statistics collection is ongoing.

More information about using AWA commands can be found in the “Data ONTAP 8.2.1 Commands: Manual Page Reference” documents.

NetApp recommends running AWA for several hours when the workload on the target aggregate is high. If it is difficult to determine when a peak workload occurs, AWA should be run for at least two to three days. Only up to one week of AWA data is retained, so NetApp recommends running AWA for no longer than one week.

The AWA summary report is designed to be easy to understand. The report is displayed on the system console and it is not logged in a file for later access. Here is a sample report:

```
### FP AWA Stats ###

                  Version 1
            Layout Version 1
                CM Version 1

Basic Information

                Aggregate aggr1
              Current-time Sat Aug 17 11:42:40 PDT 2013
                Start-time Sat Aug 17 07:46:58 PDT 2013
      Total runtime (sec) 14138
    Interval length (sec) 600
          Total intervals 24
        In-core Intervals 1024

Summary of the past 24 intervals
                              max
          Read Throughput  139.251 MB/sec
         Write Throughput  347.559 MB/sec
       Cacheable Read (%)   50.000 %
      Cacheable Write (%)   30.000 %
     Projected Cache Size  640.000 GB
   Projected Read Offload   45.000 %
  Projected Write Offload   28.000 %

Summary Cache Hit Rate vs. Cache Size

       Size        20%        40%        60%        80%       100%
   Read Hit      2.000      5.000     11.000     30.000     45.000
  Write Hit      7.000     12.000     18.000     22.000     28.000

The entire results and output of Automated Workload Analyzer (AWA) are
estimates. The format, syntax, CLI, results and output of AWA may change in
future Data ONTAP releases. AWA reports the projected cache size in capacity.
It does not make recommendations regarding the number of data SSDs required.
Please follow the guidelines for configuring and deploying Flash Pool; that
are provided in tools and collateral documents. These include verifying the
platform cache size maximums and minimum number and maximum number of data
SSDs.

### FP AWA Stats End ###
```

The projected read and write offload percentages estimate how much of the workload that is currently handled by HDDs within the aggregate could be serviced by the projected Flash Pool cache.

6.2 Sizing Flash Pool Cache with Predictive Cache Statistics

Predictive Cache Statistics (PCS) is designed to estimate the performance benefit of adding Flash Cache modules to a controller that has neither Flash Cache nor Flash Pool installed. PCS can be used to estimate Flash Pool cache size, however the methodology requires modification to do so.

PCS considers only the benefit of caching data for read requests. Therefore, the effectiveness of PCS in estimating the benefit of converting an HDD-only aggregate to a Flash Pool aggregate is limited. PCS can be used to get a rough estimate of the read cache benefit of adding Flash Pool to a system that isn't

configured with Flash Pool or Flash Cache if all the aggregates will be converted to Flash Pool aggregates, or if the aggregates that are not under consideration for conversion to Flash Pool are known to contain volumes that have only sequential read-write workloads. To estimate Flash Pool benefits, follow these guidelines when using PCS:

- PCS can be used only on a system that has neither Flash Cache nor Flash Pool configured.
- The recommended Flash Pool cache is 33% larger than the optimal cache size that PCS projects.

More information about how to use PCS is available in TR-3801: “Introduction to Predictive Cache Statistics.”

6.3 Workload Analysis with Releases Earlier than Data ONTAP 8.2.1

For releases earlier than 8.2.1, AWA is not available for workload analysis. Workload analysis can be performed with earlier releases by using statistics that are available only by using diagnostic administration access, also called *diagnostic mode*.

Note: Diagnostic mode should be used only with the supervision of NetApp Customer Support Services or field technical personnel.

Two statistics counters help determine the mix of sequential and random read workloads, respectively, on the system:

- `stats show readahead:readahead:seq_read_req`
- `stats show readahead:readahead:rand_read_req`

Six additional statistics counters are used to understand how much read I/O each system resource is servicing:

- `stats show wafl:wafl:read_io_type.cache` (system memory)
- `stats show wafl:wafl:read_io_type.ext_cache` (Flash Cache)
- `stats show wafl:wafl:read_io_type.disk` (HDD aggregates)
- `stats show wafl:wafl:read_io_type.bamboo_ssd` (SSD aggregates)
- `stats show wafl:wafl:read_io_type.hya_hdd` (Flash Pool HDDs)
- `stats show wafl:wafl:read_io_type.hya_cache` (Flash Pool SSD cache)

For a system that has neither Flash Cache nor Flash Pool configured, if most of the read workload is serviced from system memory, then adding Flash Pool (or Flash Cache, for that matter) provides little or no improvement in overall system performance (response time and throughput). In contrast, if a large fraction of read requests is being handled by disk drives (that is, HDD aggregates), then adding either Flash Cache or Flash Pool may improve performance, if the random read mix is substantial (based on the read-ahead counters).

The read-write mix of a workload can be determined by using the protocol-specific counters for the number of operations serviced by the system. For example, `show stats nfsv3:nfs:nfsv3_read_ops` and `show stats nfsv3:nfs:nfsv3_write_ops` show total the NFS system read and write operations, respectively.

If the protocol-specific counters indicate that the mix of random writes is high, it is important to understand the block/transfer sizes that are being used. Flash Pool caches repeat random overwrites that are 16KB or smaller in size. The protocol-specific “write size” histogram can be used to display the average block/transfer size on the system. Using NFS as an example again, the counter to use is `show stats nfsv3:nfs:nfsv3_write_size_histo`.

7 Conclusion

Configuring a NetApp Data ONTAP storage system with Flash Pool can significantly reduce read latencies, and will often increase system throughput, for both block- and file-based workloads, by offloading random read and random write I/O from HDDs to faster SSDs. The latency and throughput improvements can produce better application performance and result in lower-cost storage system deployments.

Version History

Version	Date	Document Version History
Version 2.1	March 2014	Skip Shapiro: Updated for Data ONTAP 8.2.1 enhancements, and addition of new platform and SSD hardware.
Version 2.0	April 2013	Skip Shapiro: Updated to incorporate enhancements with Data ONTAP 8.2 and support for additional SSD models.
Version 1.0	May 2012	Paul Updike: Initial version of this technical report.

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®

NetApp
www.netapp.com