FlexCache write-back

ONTAP 9

NetApp
January 23, 2026

Table of Contents

FlexCache write-back

Learn about ONTAP FlexCache write-back

Introduced in ONTAP 9.15.1, FlexCache write-back is an alternate mode of operation for writing at a cache. Write-back allows the write to be committed to stable storage at the cache and acknowledged to the client without waiting for the data to make it to the origin. The data is asynchronously flushed back to the origin. The result is a globally distributed file system that enables writes to perform at near-local speeds for specific workloads and environments, offering significant performance benefits.

> ONTAP 9.12.1 introduced a write-back feature as a public preview. This is referred to as write-back version 1 (wbv1) and shouldn't be thought of as the same as write-back in ONTAP 9.15.1, which is referred to as write-back version 2 (wbv2).

Write-back vs write-around

Since FlexCache was introduced in ONTAP 9.5, it has been a read-writable cache; however, it operated in write-around mode. Writes at the cache were shipped to the origin to be committed to stable storage. After the origin successfully committed the write to stable storage, it acknowledged the write to the cache. The cache would then acknowledge the write to the client. This made every write incur the penalty of traversing the network between the cache and origin. FlexCache write-back changes this.

> After upgrading to ONTAP 9.15.1, you can convert a traditional write-around cache to a write-back cache, and, if necessary, back to write-around. This can, however, make reading diagnostic logs harder should a problem arise.

	Write-around	Write-back
ONTAP Version	9.6+	9.15.1+
Use case	Read-heavy workload	Write-heavy workload
Data committed at	Origin	Cache
Client experience	WAN-like	LAN-like
Limits	100 per origin	10 per origin
CAP Theorem	Available and tolerant to partition	Available and consistent

FlexCache write-back terminology

Understand key concepts and terms working with FlexCache write-back.

Term	Definition
Dirty data	Data that has been committed to stable storage at the cache, but has not been flushed to the origin.

Term	Definition
Exclusive Lock Delegation (XLD)	A protocol-level lock authority granted on a per-file basis to a cache. This authority allows the cache to hand out exclusive write locks to clients without contacting the origin.
Shared Lock Delegation (SLD)	A protocol-level lock authority granted on a per-file basis to a cache. This authority allows the cache to hand out shared read locks to clients without contacting the origin.
Write-back	A mode of FlexCache operation where writes to a cache are committed to stable storage at that cache and immediately acknowledged to the client. Data is asynchronously written back to the origin.
Write-around	A mode of FlexCache operation where writes to a cache are forwarded to the origin to be committed to stable storage. Once committed, the origin will acknowledge the write to the cache, and the cache will acknowledge the write to the client.
Dirty Data Record System (DDRS)	A proprietary mechanism that keeps track of the dirty data in a write-back-enabled cache on a per-file basis.
Origin	A FlexGroup or FlexVol that contains the source data for all FlexCache cache volumes. It is the single source of truth, orchestrates locking, and ensures 100% data consistency, currency, and coherency.
Cache	A FlexGroup that is a sparse cache volume of the FlexCache origin.

Consistent, current, and coherent

FlexCache is NetApp's solution to having the right data, everywhere, every time. FlexCache is 100% consistent, current, and coherent 100% of the time:

- **Consistent:** The data is the same wherever it is accessed.
- **Current:** The data is always up-to-date.
- **Coherent:** The data is correct/uncorrupted.

ONTAP FlexCache write-back guidelines

FlexCache write-back involves many complex interactions between the origin and caches. For optimal performance, you should ensure your environment follows these guidelines. These guidelines are based on the latest major ONTAP version (ONTAP 9.17.1.) available at the time of content creation.

As a best practice, test your production workload in a non-production environment. This is even more important if you are implementing FlexCache write-back outside of these guidelines.

The following guidelines are well-tested internally at NetApp. It is ***strongly*** recommended you stay within them. If you do not, unexpected behavior could occur.

- Significant enhancements for FlexCache write-back were introduced in ONTAP 9.17.1P1. It is ***strongly***

advised you run the current recommended release after 9.17.1P1 at both the origin and cache clusters. If you are unable to run 9.17.1 codeline, the latest P release of 9.16.1 is the next suggested release. ONTAP 9.15.1 does not have all the necessary fixes and improvements for FlexCache write-back, and is not recommended for production workloads.

- In its current iteration, FlexCache write-back caches should be configured with a single constituent for the entire FlexCache volume. Multi-constituent FlexCaches can result in unwanted evictions of data from the cache.
- Testing has been executed for files smaller than 100GB and WAN round-trip times between the cache and origin not exceeding 200ms. Any workloads outside of these limits might result in unexpected performance characteristics.
- Writing to SMB alternate data streams causes the main file to be evicted from the cache. All dirty data for the main file needs to be flushed to the origin before any other operations can take place on that file. The alternate data stream is also forwarded to the origin.
- Renaming a file causes the file to be evicted from the cache. All dirty data for the file needs to be flushed to the origin before any other operations can take place on that file.
- At this time, the only attributes that can be changed or set on a file on the write-back-enabled FlexCache volume are:
  - Timestamps
  - Mode bits
  - NT ACLs
  - Owner
  - Group
  - Size

    Any other attributes that are changed or set are forwarded to origin which might result in evicting the file from the cache. If you require other attributes to be changed or set at the cache, ask your account team to open a PVR.
- Snapshots taken at the origin cause recalling all outstanding dirty data from every write-back-enabled cache associated with that origin volume. This might require multiple retries of the operation if there is significant write-back activity in progress, as evicts of those dirty files might take some time.
- SMB Opportunistic Locks (Oplocks) for writes are not supported on write-back-enabled FlexCache volumes.
- The origin must remain under 80% full. Cache volumes are not granted exclusive lock delegations if there isn't at least 20% space remaining in the origin volume. Calls to a write-back-enabled cache are forwarded to the origin in this situation. This helps prevent running out of space at the origin, which would result in leaving dirty data orphaned at a write-back-enabled cache.
- Low bandwidth and/or lossy intercluster networks can have a significant negative effect on FlexCache write-back performance. While there isn't a specific bandwidth requirement, as it is highly dependent on your workload, it is ***strongly*** recommended you ensure the health of the intercluster link between the cache(s) and origin.

ONTAP FlexCache write-back architecture

FlexCache was designed with strong consistency in mind, including both modes of write operation: write-back and write-around. Both the traditional write-around mode of operation and the new write-back mode of operation introduced in ONTAP 9.15.1

guarantee that the data accessed will always be 100% consistent, current, and coherent.

The following concepts detail how FlexCache write-back operates.

Delegations

Lock delegations and data delegations helps FlexCache keep both write-back and write-around caches data consistent, coherent, and current. The origin orchestrates both delegations.

Lock delegations

A lock delegation is a protocol-level lock authority the origin grants on a per-file basis to a cache to issue protocol locks to clients as needed. These include exclusive lock delegations (XLD) and shared lock delegations (SLD).

XLD and write-back

To ensure ONTAP never has to reconcile a conflicting write, an XLD is granted to a cache where a client requests to write to a file. Importantly, only one XLD can exist for any file at any time, meaning there never will be more than one writer to a file at a time.

When the request to write to a file comes into a write-back enabled cache, the following steps take place:

1. The cache checks if it already has an XLD for the requested file. If so, it will grant the write lock to the client as long as another client isn't writing to the file at the cache. If the cache doesn't have an XLD for the requested file, it will request one from the origin. This is a proprietary call that traverses the intercluster network.
2. Upon receiving the XLD request from the cache, the origin will check if there is an outstanding XLD for the file at another cache. If so, it will recall that file's XLD, which triggers a flush of any dirty data from that cache back to the origin.
3. Once the dirty data from that cache is flushed back and committed to stable storage at the origin, the origin will grant the XLD for the file to the requesting cache.
4. Once the file's XLD is received, the cache grants the lock to the client, and the write commences.

A high-level sequence diagram covering some of these steps is covered in the Write-back sequence diagram.

From a client perspective, all locking will work as if it were writing to a standard FlexVol or FlexGroup with a potential small delay when the write lock is requested.

In it's current iteration, if a write-back enabled cache holds the XLD for a file, ONTAP will block **any** access to that file at other caches, including `READ` operations.

> There is a limit of 170 XLDs per origin constituent.

Data delegations

A data delegation is a per-file guarantee given to a cache by the origin that the data cached for that file is up-to-date. As long as the cache has a data delegation for a file, it can serve the cached data for that file to the client without having to contact the origin. If the cache doesn't have a data delegation for the file, it must contact the origin to receive the data requested by the client.

In write-back mode, a file's data delegation is revoked if an XLD is taken for that file at another cache or the origin. This effectively fences off the file from clients at all other caches and the origin, even for reads. This is a

trade off that must be made to ensure old data is never accessed.

Reads at a write-back-enabled cache generally operate like reads at a write-around cache. In both write-around and write-back-enabled caches, there could be an initial `READ` performance hit when the requested file has an exclusive write lock at a write-back-enabled cache other than where the read is issued. The XLD has to be revoked, and the dirty data must be committed to the origin before the read at the other cache can be serviced.

Tracking dirty data

Write-back from cache to origin happens asynchronously. This means that dirty data isn't immediately written back to the origin. ONTAP employs a dirty data record system to keep track of dirty data per file. Each dirty data record (DDR) represents approximately 20MB of dirty data for a particular file. When a file is actively being written, ONTAP will start flushing dirty data back after two DDRs have been filled and the third DDR is being written. This results in approximately 40MB of dirty data remaining in a cache during writes. For stateful protocols (NFSv4.x, SMB), the remaining 40MB of data will be flushed back to the origin when the file is closed. For stateless protocols (NFSv3), the 40MB of data will be flushed back when either access to the file is requested at a different cache or after the file is idle for two or more minutes, up to a maximum of five minutes. For more information on timer-triggered or space-triggered dirty data flushing, see Cache scrubbers.

In addition to the DDRs and scrubbers, some front-end NAS operations also trigger the flushing of all dirty data for a file:

- `SETATTR`
  - `SETATTR`s that modify only mtime, atime, and/or ctime can be processed at the cache, avoiding the penalty of the WAN.
- `CLOSE`
- `OPEN` at another cache
- `READ` at another cache
- `READDIR` at another cache
- `READDIRPLUS` at another cache
- `WRITE` at another cache

Disconnected mode

When an XLD for a file is held at a write-around cache and that cache gets disconnected from the origin, reads for that file are still allowed at the other caches and origin. This behavior differs when an XLD is held by a write-back-enabled cache. In this case, if the cache is disconnected, reads to the file will hang everywhere. This helps ensure 100% consistency, currency, and coherence are maintained. The reads are allowed in write-around mode because the origin is guaranteed to have all of the data available that has been write-acknowledged to the client. In write-back mode during a disconnect, the origin can not guarantee that all of the data written to and acknowledged by the write-back-enabled cache made it to the origin before the disconnect occurred.

In the event a cache with an XLD for a file is disconnected for an extended period of time, a system administrator can manually revoke the XLD at the origin. This will allow IO to the file to resume at the surviving caches and the origin.

> Manually revoking the XLD will result in the loss of any dirty data for the file at the disconnected cache. Manually revoking an XLD should only be done in the event of a catastrophic disruption between the cache and origin.

Cache scrubbers

There are scrubbers in ONTAP that run in response to specific events, such as a timer expiring or space thresholds being breached. The scrubbers take an exclusive lock on the file being scrubbed, effectively freezing IO to that file until the scrub completes.

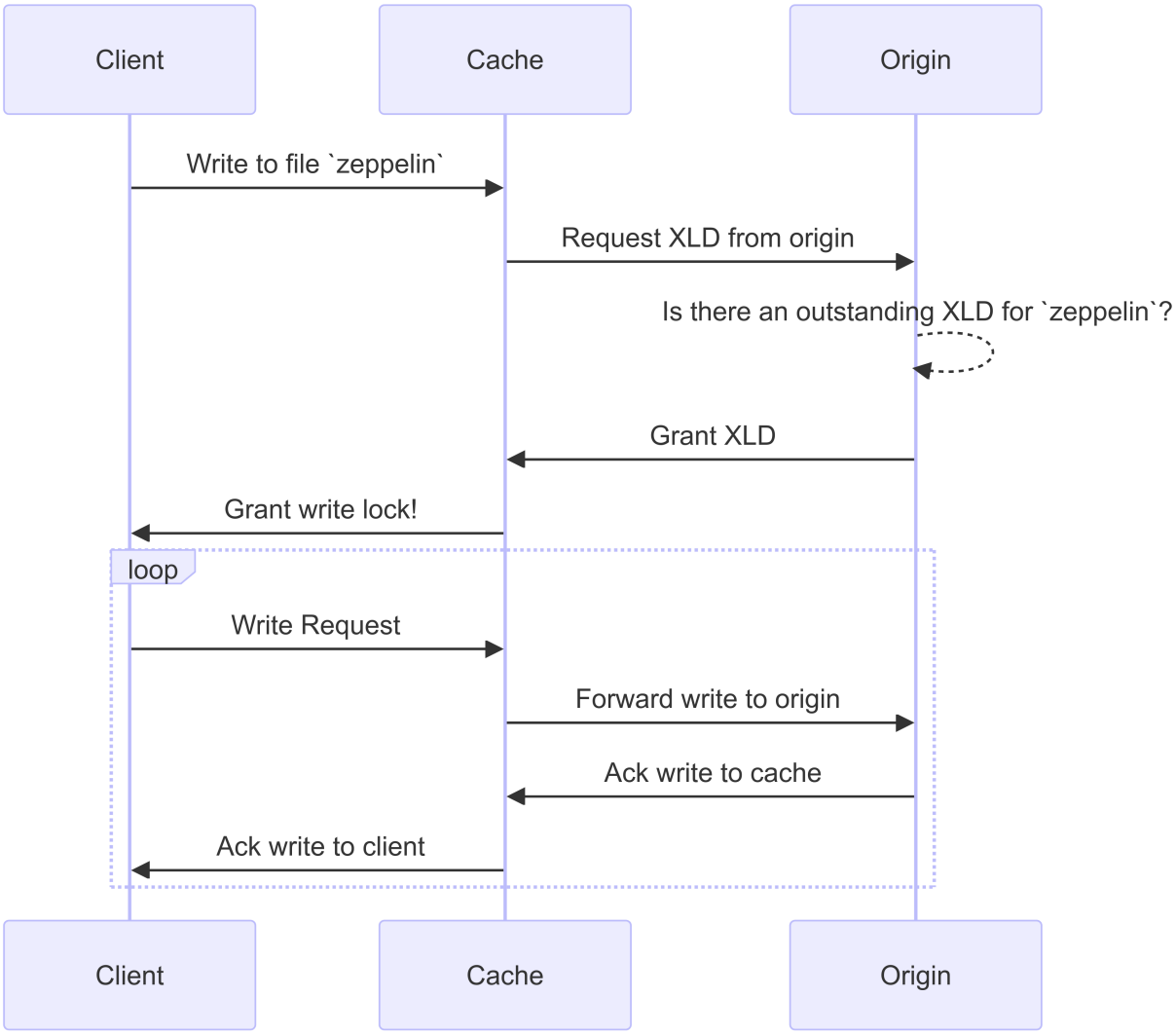
Scrubbers include:

- **mtime-based scrubber on the cache:** This scrubber starts every five minutes and scrubs any file sitting unmodified for two minutes. If any dirty data for the file is still in the cache, IO to that file is quiesced and write-back is triggered. IO will resume after the write-back is complete.
- **mtime-based scrubber on origin:** Much like the mtime-based scrubber at the cache, this also runs every five minutes. However, it scrubs any file sitting unmodified for 15 minutes, recalling the inode's delegation. This scrubber doesn't initiate any write-back.
- **RW limit-based scrubber on origin:** ONTAP monitors how many RW lock delegations are handed out per origin constituent. If this number surpasses 170, ONTAP starts scrubbing write lock delegations on a least-recently-used (LRU) basis.
- **Space-based scrubber on the cache:** If a FlexCache volume reaches 90% full, the cache is scrubbed, evicting on an LRU basis.
- **Space-based scrubber on the origin:** If a FlexCache origin volume reaches 90% full, the cache is scrubbed, evicting on an LRU basis.

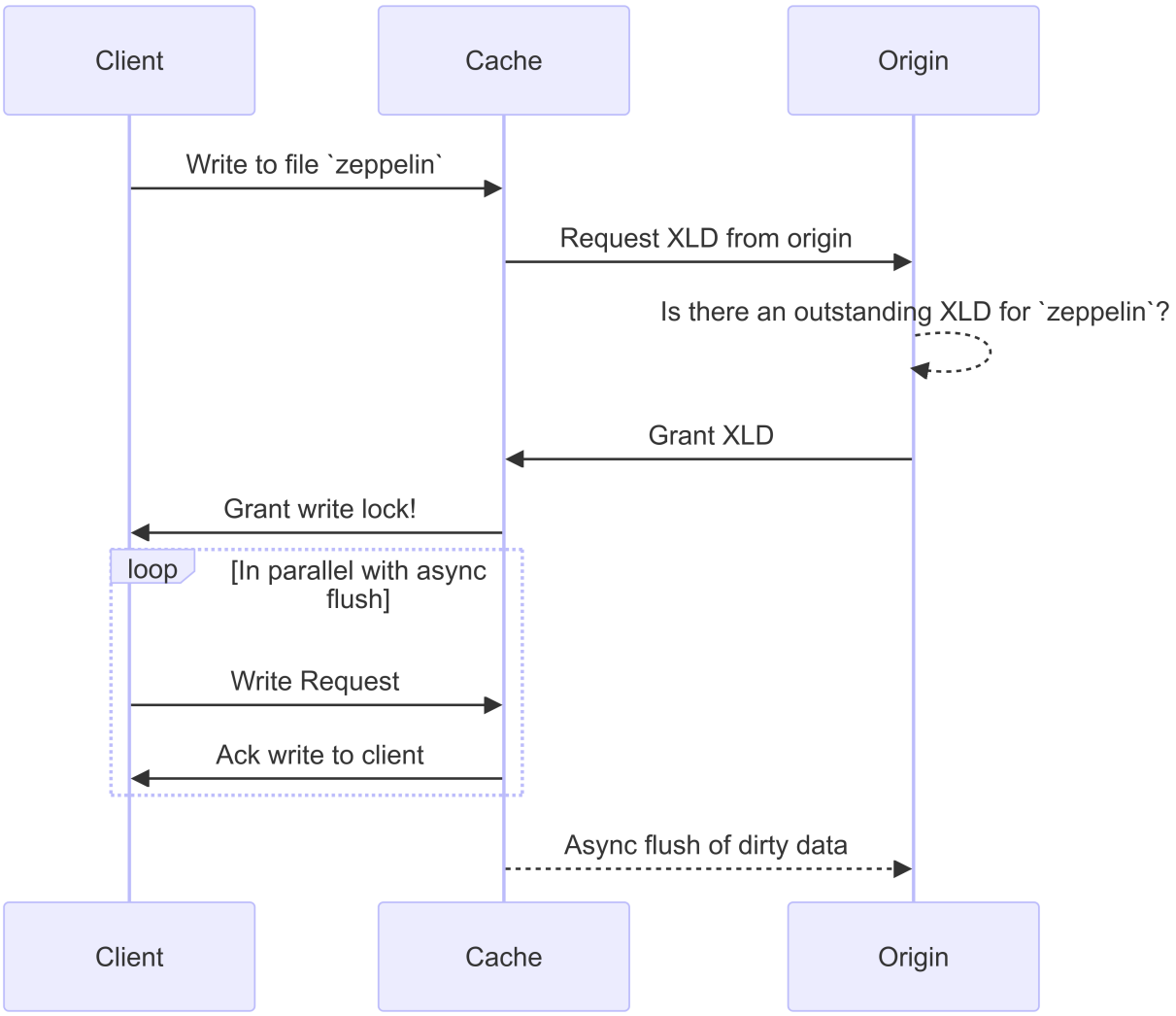
Sequence diagrams

These sequence diagrams depict the difference in write acknowledgements between write-around and write-back mode.

Write-around


Client
Cache
Origin
Write to file `zeppelin`
Request XLD from origin
Is there an outstanding XLD for `zeppelin`?
Grant XLD
Grant write lock!
loop
Write Request
Forward write to origin
Ack write to cache
Ack write to client
Client
Cache
Origin

Write-back



ONTAP FlexCache write-back use cases

These are write profiles best suited for a write-back-enabled FlexCache. You should test your workload to see if write-back or write-around provides the best performance.

> Write-back is not a replacement for write-around. Although write-back is designed with write-heavy workloads, write-around is still the better choice for many workloads.

Target workloads

File size

File size is less important than the number of writes issued between the `OPEN` and `CLOSE` calls for a file. Small files inherently have fewer `WRITE` calls, making them less ideal for write-back. Large files might have more writes between `OPEN` and `CLOSE` calls, but this isn't guaranteed.

Refer to the FlexCache write-back guidelines page for the most current recommendations regarding max file size.

Write size

When writing from a client, other modifying NAS calls are involved other than write calls. These include, but are

not limited to:

- `CREATE`
- `OPEN`
- `CLOSE`
- `SETATTR`
- `SET_INFO`

`SETATTR` and `SET_INFO` calls that set `mtime`, `atime`, `ctime`, `owner`, `group`, or `size` are processed at the cache. The rest of these calls must be processed at the origin and trigger a write-back of any dirty data accumulated at the write-back-enabled cache for the file being operated on. IO to the file will be quiesced until the write-back is complete.

Knowing that these calls must traverse the WAN helps you to identify workloads suited for write-back. Generally, the more writes that can be done between `OPEN` and `CLOSE` calls without one of the other calls listed above being issued, the better the performance gain write-back provides.

Read-after-write

Read-after-write workloads have historically performed poorly at FlexCache. This is due to the write-around mode of operation before 9.15.1. The `WRITE` call to the file has to be committed at the origin, and the subsequent `READ` call would have to pull the data back to the cache. This results in both operations incurring the penalty of the WAN. Therefore, read-after-write workloads are discouraged for FlexCache in write-around mode. With the introduction of write-back in 9.15.1, data is now committed at the cache, and can immediately be read from the cache, eliminating the WAN penalty. If your workload includes read-after-write at FlexCache volumes, you should configure the cache to operate in write-back mode.

> If read-after-write is a critical part of your workload, you should configure your cache to operate in write-back mode.

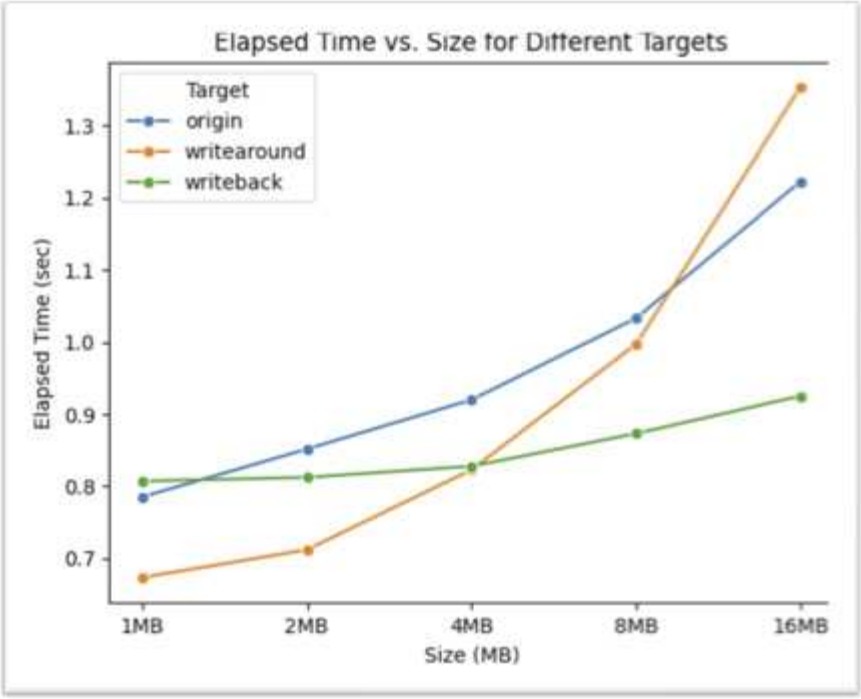
Write-after-write

When a file accumulates dirty data in a cache, the cache asynchronously writes the data back to the origin. This naturally leads to times when the client closes the file with dirty data still waiting to be flushed back to origin. If another open or write comes in for the file that was just closed and still has dirty data, the write will be suspended until all the dirty data has been flushed to origin.

Latency considerations

When FlexCache operates in write-back mode, it becomes more beneficial to NAS clients as latency increases. There is a point, however, at which the overhead of write-back outweighs the advantages gained in low-latency environments. In some NetApp tests, write-back benefits started around a minimum latency between cache and origin of 8ms. This latency varies with workload, so be sure to test to know your workload's point-of-return.

The following graph shows the point-of-return for write-back in NetApp lab tests. The `x` axis is the file-size, and the `y` axis is the elapsed time. The test used NFSv3, mounting with an `rsize` and `wsize` of 256KB, and 64ms of WAN latency. This test was performed using a small ONTAP Select instance for both the cache and origin, and a single threaded-write operation. Your results might vary.



Write-back should not be used for intracluster caching. Intracluster caching occurs when the origin and cache are in the same cluster.

ONTAP FlexCache write-back prerequisites

Before you deploy FlexCache in write-back mode, ensure you have met these performance, software, licensing, and system configuration requirements.

CPU and Memory

It is ***strongly recommended*** that each origin cluster node have at least 128GB of RAM and 20 CPUs to absorb the write-back messages initiated by write-back enabled caches. This is the equivalent of an A400 or greater. If the origin cluster serves as the origin to multiple write-back enabled FlexCaches, it will require more CPU and RAM.

Using an undersized origin for your workload can have profound impacts on performance at the write-back-enabled cache or the origin.

ONTAP version

- The origin ***must*** be running ONTAP 9.15.1 or later.
- Any caching cluster that needs to operate in write-back mode ***must*** be running ONTAP 9.15.1 or later.
- Any caching cluster that does not need to operate in write-back mode can run any generally supported ONTAP version.

Licensing

FlexCache, including the write-back mode of operation, is included with your ONTAP purchase. No extra license is required.

Peering

- The origin and cache clusters must be cluster peered
- The server virtual machines (SVMs) on the origin and cache cluster must be vserver peered with the FlexCache option.

> You do not need to peer a cache cluster to another cache cluster. There is also no need to peer a cache SVM to another cache SVM.

ONTAP FlexCache write-back interoperability

Understand these interoperability considerations when deploying FlexCache in write-back mode.

ONTAP version

To use the write-back mode of operation, both the cache and origin **must** be running ONTAP 9.15.1 or later.

> Clusters where a write-back-enabled cache is unnecessary can run earlier versions of ONTAP, but that cluster can only operate in write-around mode.

You can have a mix of ONTAP versions in your environment.

Table 1. Mixed cluster versions example 1

Cluster	ONTAP version	Write-back supported?
Origin	ONTAP 9.15.1	N/A †
Cluster 1	ONTAP 9.15.1	Yes
Cluster 2	ONTAP 9.14.1	No

Table 2. Mixed cluster versions example 2

Cluster	ONTAP version	Write-back supported?
Origin	ONTAP 9.14.1	N/A †
Cluster 1	ONTAP 9.15.1	No
Cluster 2	ONTAP 9.15.1	No

† *Origins aren't a cache, so neither write-back nor write-around support is applicable.*

> In Mixed cluster versions example 2, neither cluster can enable write-back mode because the origin is not running ONTAP 9.15.1 or later, which is a strict requirement.

Client interoperability

Any client generally supported by ONTAP can access a FlexCache volume regardless of whether it is operating in write-around or write-back mode. For an up-to-date list of supported clients, refer to NetApp's interoperability matrix.

Although the client version doesn't matter specifically, the client must be new enough to support NFSv3, NFSv4.0, NFSv4.1, SMB2.x, or SMB3.x. SMB1 and NFSv2 are deprecated protocols and are not supported.

Write-back and write-around

As seen in Mixed cluster versions example 1, FlexCache operating in write-back mode can co-exist with caches operating in write-around mode. It is advised to compare write-around against write-back with your specific workload.

> If the performance for a workload is the same between write-back and write-around, use write-around.

ONTAP feature interoperability

For the most up-to-date list of FlexCache feature interoperability, refer to the supported and unsupported features for FlexCache volumes.

Enable and manage ONTAP FlexCache write-back

Beginning with ONTAP 9.15.1, you can enable FlexCache write-back mode on FlexCache volumes to provide better performance for edge computing environments and caches with write-heavy workloads. You can also determine whether write-back is enabled on a FlexCache volume or disable write-back on the volume when necessary.

When write-back is enabled on the cache volume, write requests are sent to the local cache rather than to the origin volume.

Before you begin

You must be in advanced privilege mode.

Create a new FlexCache volume with write-back enabled

Steps

You can create a new FlexCache volume with write-back enabled by using ONTAP System Manager or the ONTAP CLI.

System Manager

1. If the FlexCache volume is on a different cluster than the origin volume, create a cluster peer relationship:
   a. On the local cluster, click **Protection > Overview**.
   b. Expand **Intercluster Settings**, click **Add Network Interfaces**, and add intercluster interfaces to the cluster.

      Repeat this on the remote cluster.

   c. On the remote cluster, click **Protection > Overview**. Click ⋮ in the Cluster Peers section and click **Generate Passphrase**.
   d. Copy the generated passphrase and paste it in the local cluster.
   e. On the local cluster, under Cluster Peers, click **Peer Clusters** and peer the local and remote clusters.
2. If the FlexCache volume is on a different cluster than the origin volume, create an SVM peer relationship:

   Under **Storage VM Peers**, click ⋮ and then **Peer Storage VMs** to peer the storage VMs.

   If the FlexCache volume is on the same cluster, you cannot create an SVM peer relationship using System Manager.

3. Select **Storage > Volumes**.
4. Select **Add**.
5. Select **More Options** and then select **Add as cache for a remote volume**.
6. Select **Enable FlexCache write-back**.

CLI

1. If the FlexCache volume to be created is in a different cluster, create a cluster peer relationship:
   a. On the destination cluster, create a peer relationship with the data protection source cluster:

      ```
      cluster peer create -generate-passphrase -offer-expiration
      MM/DD/YYYY HH:MM:SS|1...7days|1...168hours -peer-addrs
      <peer_LIF_IPs> -initial-allowed-vserver-peers <svm_name>,..|*
      -ipspace <ipspace_name>
      ```

      Beginning with ONTAP 9.6, TLS encryption is enabled by default when creating a cluster peer relationship. TLS encryption is supported for the intercluster communication between the origin and FlexCache volumes. You can also disable TLS encryption for the cluster peer relationship, if required.

```
cluster02::> cluster peer create -generate-passphrase -offer
-expiration 2days -initial-allowed-vserver-peers *

                     Passphrase: UCa+6lRVICXeL/gq1WrK7ShR
                Expiration Time: 6/7/2017 08:16:10 EST
  Initial Allowed Vserver Peers: *
            Intercluster LIF IP: 192.140.112.101
              Peer Cluster Name: Clus_7ShR (temporary generated)

Warning: make a note of the passphrase - it cannot be displayed
again.
```

b. On the source cluster, authenticate the source cluster to the destination cluster:

```
cluster peer create -peer-addrs <peer_LIF_IPs> -ipspace <ipspace>
```

```
cluster01::> cluster peer create -peer-addrs
192.140.112.101,192.140.112.102

Notice: Use a generated passphrase or choose a passphrase of 8 or
more characters.
        To ensure the authenticity of the peering relationship,
use a phrase or sequence of characters that would be hard to
guess.

Enter the passphrase:
Confirm the passphrase:

Clusters cluster02 and cluster01 are peered.
```

2. If the FlexCache volume is in a different SVM than that of the origin volume, create an SVM peer relationship with `flexcache` as the application:

   a. If the SVM is in a different cluster, create an SVM permission for the peering SVMs:

```
vserver peer permission create -peer-cluster <cluster_name>
-vserver <svm-name> -applications flexcache
```

   The following example illustrates how to create an SVM peer permission that applies for all of the local SVMs:

```
cluster1::> vserver peer permission create -peer-cluster cluster2
-vserver "*" -applications flexcache

Warning: This Vserver peer permission applies to all local
Vservers. After that no explict
"vserver peer accept" command required for Vserver peer
relationship creation request
from peer cluster "cluster2" with any of the local Vservers. Do
you want to continue? {y|n}: y
```

b. Create the SVM peer relationship:

```
vserver peer create -vserver <local_SVM> -peer-vserver
<remote_SVM> -peer-cluster <cluster_name> -applications flexcache
```

3. Create a FlexCache volume with write-back enabled:

```
volume flexcache create -vserver <cache_vserver_name> -volume
<cache_flexgroup_name> -aggr-list <list_of_aggregates> -origin
-volume <origin flexgroup> -origin-vserver <origin_vserver name>
-junction-path <junction_path> -is-writeback-enabled true
```

Enable FlexCache write-back on an existing FlexCache volume

You can enable FlexCache write-back on an existing FlexCache volume using ONTAP System Manager or the ONTAP CLI.

System Manager

1. Select **Storage > Volumes** and select an existing FlexCache volume.
2. On the volume's Overview page, click **Edit** in the upper right corner.
3. In the **Edit Volume** window, select **Enable FlexCache write-back**.

CLI

1. Enable write-back on an existing FlexCache volume:

```
volume flexcache config modify -volume <cache_flexgroup_name> -is
-writeback-enabled true
```

Check if FlexCache write-back is enabled

Steps

You can use System Manager or the ONTAP CLI to determine whether FlexCache write-back is enabled.

System Manager

1. Select **Storage > Volumes** and select a volume.
2. In the volume **Overview**, locate **FlexCache details** and check if FlexCache write-back is set to **Enabled** on the FlexCache volume.

CLI

1. Check if FlexCache write-back is enabled:

```
volume flexcache config show -volume <cache_flexgroup_name> -fields
is-writeback-enabled
```

Disable write-back on a FlexCache volume

Before you can delete a FlexCache volume you need to disable FlexCache write-back.

Steps

You can use System Manager or the ONTAP CLI to disable FlexCache write-back.

System Manager

1. Select **Storage > Volumes** and select an existing FlexCache volume that has FlexCache write-back enabled.
2. On the volume's Overview page, click **Edit** in the upper right corner.
3. In the **Edit Volume** window, deselect **Enable FlexCache write-back**.

CLI

1. Disable write-back:

```
volume flexcache config modify -volume <cache_vol_name> -is
-writeback-enabled false
```

Frequently asked questions about ONTAP FlexCache write-back

This FAQ can help if you are looking for a quick answer to a question.

I want to use write-back. What version of ONTAP do I need to run?

Both the cache and the origin must be running ONTAP 9.15.1 or later. It is ***strongly*** recommended that you run the latest P release. Engineering is constantly improving the performance and functionality of write-back-enabled caches.

Can clients accessing the origin have an effect on clients accessing the write-back-enabled cache?

Yes. The origin has equal right to the data as any of the caches. If an operation is executed on a file that requires the file to be evicted from the cache, or a lock/data delegation to be revoked, the client at the cache might see a delay accessing the file.

Can I apply QoS to write-back-enabled FlexCaches?

Yes. Every cache and the origin can have independent QoS policies applied. This will have no direct effect on any write-back initiated intercluster traffic. Indirectly, you can slow down intercluster write-back traffic by QoS limiting the front-end traffic at the write-back-enabled cache.

Is multi-protocol NAS supported at write-back-enabled FlexCaches?

Yes. Multi-protocol is fully supported at write-back-enabled FlexCaches. Currently, NFSv4.2 and S3 are not supported by FlexCache operating in write-around or write-back mode.

Are SMB alternate data streams supported at write-back-enabled FlexCaches?

SMB alternate data streams (ADS) are supported, but not accelerated by write-back. The write to the ADS is forwarded to the origin, incurring the penalty of the WAN latency. The write also evicts the main file the ADS is a part of from the cache.

Can I switch a cache between write-around and write-back mode after it is created?

Yes. All you have to do is toggle the `is-writeback-enabled` flag in the `flexcache modify` command.

Are there bandwidth considerations I should be aware of for the intercluster link between the cache(s) and origin?

Yes. FlexCache write-back is highly dependent on the intercluster link between the cache(s) and origin. Low bandwidth and/or lossy networks can have a significant negative effect on performance. There isn't a specific bandwidth requirement, as it is highly dependent on your workload.