Technical Report

Database Storage Tiering with NetApp FabricPool

Jeffrey Steiner, BS Navyashree, Srinivas Venkat, NetApp

April 2021 | TR-4695

Abstract

This document describes the benefits and configuration options of NetApp® FabricPool® with various databases, including the Oracle relational database management system (RDBMS).

TABLE OF CONTENTS

LIST OF FIGURES

Introduction

NetApp FabricPool is an automated storage tiering feature in which active data resides on local high-performance solid-state drives (SSDs), and inactive data is tiered to low-cost object storage. It was first made available in NetApp ONTAP® 9.2 for management of read-only data and has since been enhanced in ONTAP 9.4 to also tier active data.

In a database context, you can create a single storage architecture in which the hot data remains on the local storage array. Inactive data such as archived logs, database backups, or even less active database blocks are moved to less expensive object storage.

FabricPool is integrated with ONTAP. The tiering process is fully automated with multiple policy-based management options. Apart from the performance characteristics of the object storage layer, FabricPool is transparent to application and database configuration. No architectural changes are required, and you can continue managing your database and application environment from the central ONTAP storage system.

FabricPool and NVMe

You should also consider FabricPool an important part of your Non-Volatile Memory Express (NVMe) strategy. FabricPool integrates an NVMe performance tier with an object-storage capacity tier. Only the data that truly belongs on the NVMe-based media remains on the performance tier, providing maximum return on your NVMe investment.

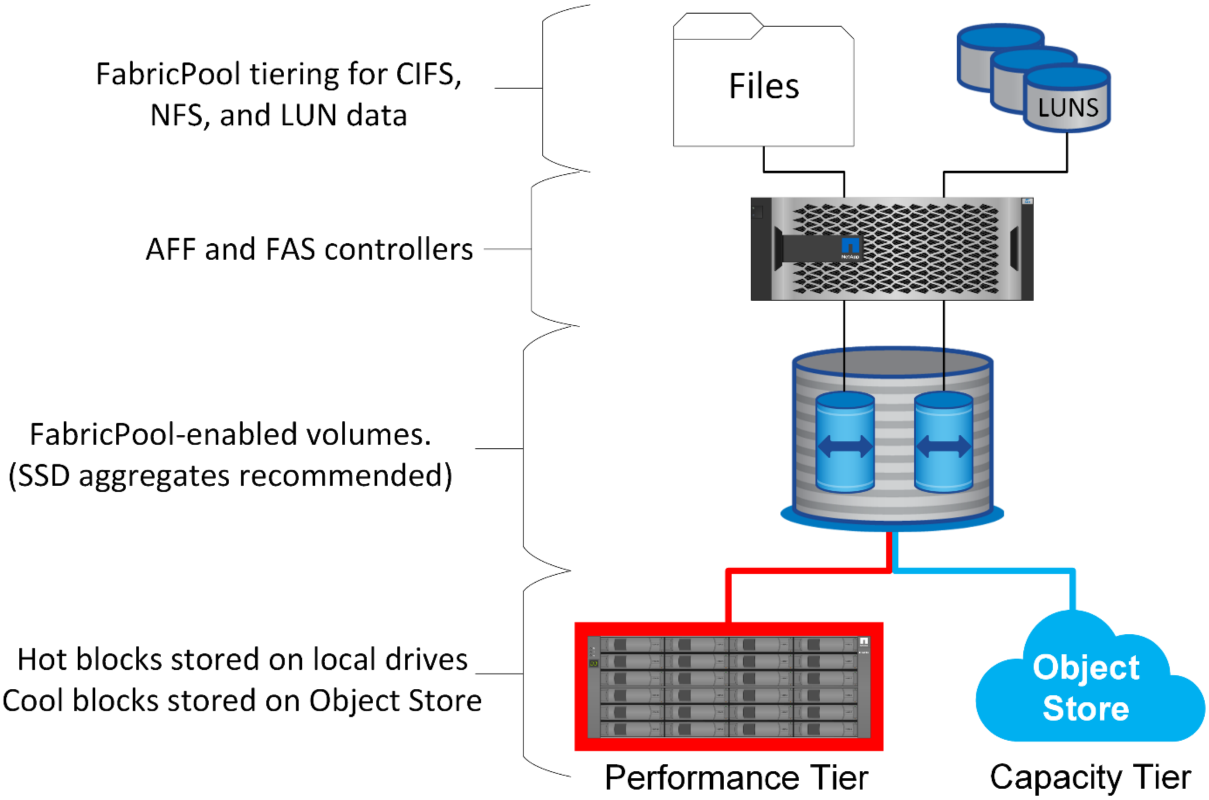
FabricPool overview

Architecture

FabricPool is a tiering technology that classifies blocks as hot or cool and places them in the most appropriate tier of storage. The performance tier is located on SSD storage and hosts the hot data blocks. The capacity tier is located on an object-store destination and hosts the cool data blocks. Object storage support includes NetApp StorageGRID®, NetApp ONTAP S3, Microsoft Azure Blob storage, Alibaba Cloud Object Storage, IBM Cloud Object Storage, Google Cloud, and Amazon Simple Storage Service (Amazon S3). Figure 1 shows the FabricPool architecture.

Multiple tiering policies are available that control how blocks are classified as hot or cool, and policies can be set on a per-volume basis and changed as required. Only the data blocks are moved between the performance and capacity tiers. The metadata that defines the LUN and file system structure always remains on the performance tier. As a result, management is centralized on ONTAP. Files and LUNs appear no different from data stored on any other ONTAP configuration. The NetApp AFF or FAS controller applies the defined policies to move data to the apprJefopriate tier.

Figure 1) FabricPool architecture.


FabricPool tiering for CIFS, NFS, and LUN data
Files
LUNS
AFF and FAS controllers
FabricPool-enabled volumes. (SSD aggregates recommended)
Hot blocks stored on local drives
Cool blocks stored on Object Store
Object Store
Performance Tier
Capacity Tier

Support

Support is included for the following configurations:

- AFF systems
- Most FAS systems capable of running ONTAP 9.2. SSD aggregates are recommended, but HDD is also supported.
- ONTAP Select. NetApp recommends using all SSD FabricPool aggregates.
- ONTAP Cloud

Object store providers

Object storage protocols use simple, HTTP or HTTPS requests for storing large numbers of data objects. Access to the object storage must be reliable because data access from ONTAP depends on prompt servicing of requests. Options include the Amazon S3 Standard and Infrequent Access options, and Microsoft Azure Hot and Cool Blob Storage, IBM Cloud, and Google Cloud. Archival options such as Amazon Glacier and Amazon Archive are not supported because the time required to retrieve data can exceed the tolerances of host operating systems and applications.

NetApp StorageGRID is also supported and is an optimal enterprise-class solution. It is a high-performance, scalable, and highly secure object storage system that can provide geographic redundancy for FabricPool data as well as other object store applications that are increasingly likely to be part of enterprise database environments.

StorageGRID can also reduce costs by avoiding the egress charges imposed by many public cloud providers for reading data back from their services.

Data and metadata

Note that the term "data" here applies to the actual data blocks, not the metadata. Only data blocks are tiered, while metadata remains local to the SSDs. In addition, the status of a block as hot or cool is only affected by reading the actual data block. Simply reading the name, timestamp, or ownership metadata of a file does not affect the location of the underlying data blocks.

Backups

Although FabricPool can significantly reduce storage footprints, it is not by itself a backup solution. NetApp WAFL® metadata always stays on the performance tier. If a catastrophic disaster destroys the performance tier, a new environment cannot be created using the data on the capacity tier because it contains no WAFL metadata.

FabricPool can, however, become part of a backup strategy. For example, FabricPool can be configured with NetApp SnapMirror® replication technology. Each half of the mirror can have its own connection to an object storage target. The result is two independent copies of the data. The primary copy consists of the blocks on the performance tier and associated blocks in the capacity tier, and the replica is a second set of performance and capacity blocks.

Tiering policies

A NetApp FlexVol® volume is the basic unit of management for ONTAP. A FlexVol volume does not consume space by itself; rather, it is just a container for files and/or LUNs. Policies, including FabricPool tiering policies, are then set on the volumes as desired.

Four policies are available in ONTAP that control how a block on the performance tier becomes a candidate to be relocated to the capacity tier.

Snapshot-only

The `snapshot-only tiering-policy` applies only to blocks that are not shared with the active file system. It essentially results in tiering of database backups. Blocks become candidates for tiering after a snapshot is created and the block is then overwritten, resulting in a block that exists only within the snapshot copy. The delay before a `snapshot-only` block is considered cool is controlled by the `tiering-minimum-cooling-days` setting for the volume. From ONTAP 9.8 and later, the range is from two to 183 days.

Most databases have low change rates, resulting in minimal savings from this policy. For example, a typical database observed on ONTAP has a change rate of less than 5% per week. Exceptions exist, but the rates are usually low. Database archive logs can occupy extensive space, but they usually continue to exist in the active file system, and thus, are not candidates for tiering under this policy.

Auto

The `auto` tiering policy extends tiering to both snapshot-specific blocks as well as blocks within the active file system. The delay before a block is considered cool is controlled by the `tiering-minimum-cooling-days` setting for the volume. From ONTAP 9.8 and later, the range is from two to 183 days.

This approach enables tiering options that are not available with the `snapshot-only` policy. For example, a data protection policy might require 90 days of archived log files to be retained. Setting a cooling period of three days results in any log files older than three days to be tiered out from the performance layer. This action frees up substantial space on the high-speed storage platform while still allowing you to view and manage the full 90 days of data. For a more details, see section 0, “Log archiving.”

None

The `none` policy prevents any additional blocks from being tiered from the storage layer, but any data still in the capacity tier remains in the capacity tier until it is read. If the block is then read, it is pulled back and placed on the performance tier.

The primary reason to use the `none` policy is to prevent blocks from being tiered, but it can be useful for changing policies over time, such as, for example, in the case that a specific database is extensively tiered to the capacity layer but an unexpected need for full performance capabilities arises. You can change the policy to prevent any additional tiering and to confirm that any blocks read back as the database becomes more active remain in the performance tier.

All

The `all` policy replaces the `backup` policy as of ONTAP 9.6. The `backup` policy applied only to data protection volumes, meaning a SnapMirror or NetApp SnapVault® destination. The `all` policy functions in the same way but is not restricted to data protection volumes.

With this policy, blocks are immediately considered cool and eligible to be tiered to the capacity layer immediately.

This policy is especially appropriate for long-term backups. You can use it as a form of hierarchical storage management (HSM). In the past, HSM was commonly used to tier the data blocks of a file to tape while keeping the file itself visible on the file system. A FabricPool volume with the `all` policy enables you to store files in a visible and manageable way yet consume nearly no space on the local storage tier.

Retrieval policies

The tiering policies control which blocks are tiered from the performance layer to the capacity layer. Retrieval policies control what happens when a block that has been tiered is read.

Default

All FabricPool volumes are initially set to `default`, which means that the behavior is controlled by the `cloud-retrieval-policy.` The exact behavior depends on the tiering policy used.

- `auto`. Only retrieves randomly read data.
- `snapshot-only`. Retrieves all sequentially or randomly read data.
- `none`. Retrieves all sequentially or randomly read data.
- `all`. Does not retrieve data from the capacity tier.

On-read

Setting `cloud-retrieval-policy` to on-read overrides the default behavior so that a read of any tiered data causes that data to be returned to the performance tier.

For example, a volume might have been lightly used for a long time under the `auto` tiering policy and most of the blocks are now tiered out. If an unexpected change in business needs requires some of the data to be repeatedly scanned in order to prepare a certain report, it might be desirable to change the `cloud-retrieval-policy` to `on-read` to make sure that **all** the data that is read is returned to the performance tier, including **both** sequentially and randomly read data. This will improve the performance of sequential I/O operations against the volume.

Promote

The behavior of the promote policy depends on the tiering policy. If the tiering policy is `auto`, then setting the `cloud-retrieval-policy` to `promote` brings back all blocks from the capacity tier on the next tiering scan.

If the tiering policy is `snapshot-only`, then the only blocks that are returned are the blocks that are associated with the active file system. Normally, this does not have any effect because the only blocks tiered under the `snapshot-only` policy are blocks associated exclusively with snapshots. There are no tiered blocks in the active file system. If, however, data on a volume is restored by a volume SnapRestore or file-clone operation from a snapshot, some of the blocks that were tiered out because they were only associated with snapshots might now be required by the active file system. It might be desirable to temporarily change the `cloud-retrieval-policy` policy to `promote` to quickly retrieve all locally required blocks.

Never

Do not retrieve blocks from the capacity tier.

Databases with FabricPool designs

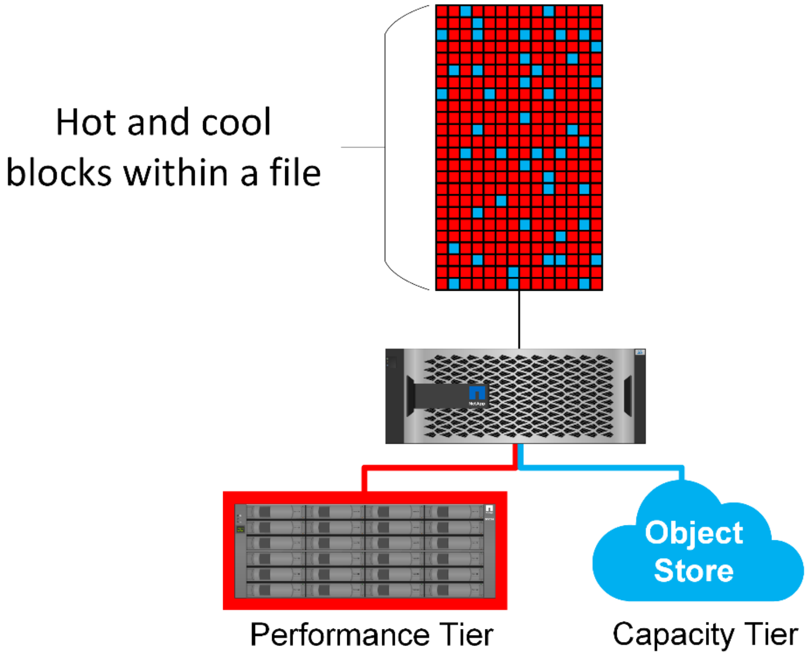
There are multiple ways to use FabricPool with database workloads and datasets. The following sections explain the basic options and are not mutually exclusive.

FabricPool and database workloads

FabricPool operates at the block level from the ONTAP point of view, but the effective results enable you to tier entire files as well as blocks within a larger file.

For example, the primary I/O pattern on a database data file is random reads and writes. FabricPool with the `auto` policy delivers a volume with active blocks that are on the SSD performance tier while the cold blocks are relocated to the capacity tier. The database itself still sees normal files at a single location, but the actual location of the blocks is as depicted in Figure 2.

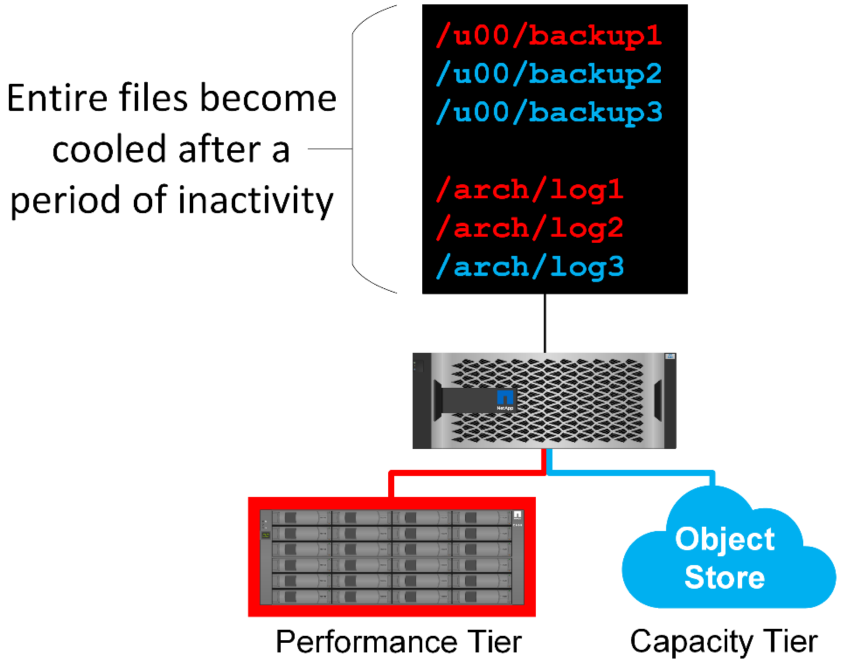
Figure 2) Sub-file block tiering.



In other cases, entire files become inactive. This includes files such as archived transaction logs or flat-file backups. These types of file are normally kept online for a certain number of days to provide high-speed access if database recovery is required. However, after a few days, they are highly unlikely to be needed again. These types of files are normally written once and are not subsequently accessed.

It's important to understand that FabricPool operates at the block level, not the file level. If datasets have files that are no longer accessed, then those entire files become tiered to the capacity layer (Figure 3).

Figure 3) Whole-file block tiering.



Note: Any type of data access resets the heat-map data. Therefore, database full-table scans and even backup activity that reads the source files prevents tiering because the required `tiering-minimum-cooling-days` threshold is never reached.

Log archiving

Perhaps the most important use for FabricPool is improving the efficiency of archived log file management. Most relational databases operate in log archival mode to deliver point-in-time recovery. Changes to the databases are committed by recording the changes in the transaction logs, and the transaction log is retained without being overwritten. In some cases, the transaction log is created as a copy of the active log file. In other cases, the active log file is closed, and the database creates a new log file in the same location but with a different name.

Recovery of the database is performed by restoring the data files to a time immediately before the required recovery point and then replaying the transaction logs until the database is at the desired state.

The result can be a requirement to retain an enormous volume of archived transaction logs. The capacity required can be many times the size of the database itself. NetApp has seen databases with over 100TB of archived logs that were kept online and available to meet regulatory requirements.

This situation creates a significant management challenge. The transaction logs can be retained on disk, but this becomes costly and wasteful, especially as databases move toward NVMe-based storage. A different tier of SATA storage can be created specifically for logs, but that approach creates an additional storage type that must be deployed and managed. Still other customers continue to use tape-based backup solutions that sweep logs to tape and remove them from the active file system. This is not only complicated, but also risky because a tape-based backup system is much more prone to problems that interrupt service.

FabricPool solves these problems by delivering a single solution with integrated tiering. The transaction logs are stored and remain accessible in their usual location.

Policies

A `tiering-minimum-cooling-days` policy of a few days retains the most recent logs, and therefore the logs most likely to be required for an urgent recovery situation, on the performance tier. The data blocks from older files are then moved to the capacity tier.

The `auto` policy is usually the most appropriate policy for a volume hosting archive log data **in the active file system.** This applies to databases such as Oracle, which perform circular overwrites of the same set of redo log files.

The `auto` policy enforces prompt tiering when the cooling threshold has been reached irrespective of whether the logs have been deleted or continue to exist in the primary file system. Storing all the potentially required logs in a single location in the active file system also simplifies management. There is no reason to search through snapshots to locate a file that needs to be restored.

Some databases, such as Microsoft SQL Server, truncate transaction log files during backup operations so that the logs are no longer in the active file system. You might save capacity by using the `snapshot-only` tiering policy, but the `auto` policy will not be useful for log data because there is rarely cooled log data in the active file system.

The `snapshot-only` policy only tiers blocks that are no longer in the active file system. Therefore, you must delete archived logs on an NFS or SMB share before the data can be tiered.

Tiering is even less efficient with a LUN configuration because deletion of a file from a LUN only removes the file references from the file system metadata. The actual blocks on the LUNs remain in place until overwritten. This might create a lengthy delay between the time a file is deleted and the time that the blocks are overwritten and become candidates for tiering.

A SnapMirror or SnapVault destination volume containing archived log files can also use a FabricPool `all` policy for the immediate tiering of all data, but the savings might not be significant. The `auto` policy normally leaves a very small amount of data on the performance tier. If the replicas are needed for recovery purposes, it would probably be better to have the recently created files available on the fastest possible tier.

Full data file tiering

Many databases include data files organized by date, and such data is generally increasingly less likely to be accessed as it ages. For example, a large billing database might contain five years of customer data, but only the most recent few months are active. You can use FabricPool to relocate older data files to the capacity tier.

Some databases, such as an Oracle database using the Information Lifecycle Management assistant, can relocate datafiles between file systems based on certain criteria. You can use such a system with FabricPool to relocate files from active data volumes to archival data volumes. For example, the primary datafiles containing the most recent three months of data might reside on SSD storage without FabricPool. After 90 days, the individual datafiles are then moved to a volume configured with a two-day cooling period to drive rapid tiering to the capacity layer.

The datafiles can also share a common FabricPool-enabled volume with a longer cooling period to make sure that required data remains available on the performance tier. There is some risk that data is relocated to the capacity tier earlier than desired, but the cooling period only applies after data is not accessed for the required period. Data does not immediately become tiered merely because the cooling period has elapsed.

Note: Any type of access to data resets the heat map data. Therefore, database full table scans and even backup activity that reads the source files prevents tiering because the required `tiering-minimum-cooling-days` threshold is never reached.

Policies

The `tiering-minimum-cooling-days` policy should be set high enough so that files that you might need remain on the performance tier. For example, a database for which the most recent 60 days of data is required with the optimal performance warrants setting the `tiering-minimum-cooling-days` period to `60`. Similar results can also be achieved based on the file access patterns. For example, if the most recent 90 days of data is required and the application is accessing that 90-day span of data, then the data will remain on the performance tier. By setting the `tiering-minimum-cooling-days` period to `2`, you get prompt tiering after the data becomes less active.

The `auto` policy is required to drive tiering of data files because only the `auto` policy affects blocks that are in the active file system.

Note: Any type of access to data resets the heat map data. Therefore, database full table scans and even backup activity that reads the source files prevents tiering because the required `tiering-minimum-cooling-days` threshold is never reached.

Database block tiering

Data files that are known to contain inactive blocks are also candidates for FabricPool tiering. For example, a supply chain management database might contain historical information that must be available if needed but is not accessed during normal operations. FabricPool can be used to selectively relocate the inactive blocks.

For example, data files running on a FabricPool volume with a `tiering-minimum-cooling-days` period of 90 days retains any blocks accessed in the preceding 90 days on the performance tier. However, anything that is not accessed for 90 days is relocated to the capacity tier. In other cases, normal application activity preserves the correct blocks on the correct tier. For example, if a database is normally used to process the previous 60 days of data on a regular basis, a much lower `tiering-minimum-cooling-days` period can be set because the natural activity of the application makes sure that blocks are not relocated prematurely.

Note: Any type of access to data resets the heat map data. Therefore, database full table scans and even backup activity that reads the source files prevents tiering because the required `tiering-minimum-cooling-days` threshold is never reached.

Policies

The `tiering-minimum-cooling-days` policy should be set high enough to retain files that might be required on the performance tier. For example, a database in which the most recent 60 days of data might be required with optimal performance would warrant setting the `tiering-minimum-cooling-days` period to 60 days. Similar results could also be achieved based on the access patterns of files. For example, if the most recent 90 days of data is required and the application is accessing that 90-day span of data, then the data would remain on the performance tier. Setting the `tiering-minimum-cooling-days` period to two days would tier the data promptly after the data becomes less active.

The `auto` policy is required to drive tiering of data file blocks because only the `auto` policy affects blocks that are in the active file system.

Local snapshot-based backups

The initial release of FabricPool targets the backup use case. The only type of blocks that you can tier are blocks that are no longer associated with data in the active file system. Therefore, you can move only the snapshot data blocks to the capacity tier.

A typical database observed on NetApp storage has approximately 5% turnover per week. There are exceptions, but in general the turnover is low. A typical snapshot-based backup approach includes backups of the data files every 6 to 24 hours. Therefore, total snapshot space consumption is approximately 15% of the total database size per week. At first, this percentage does not appear to be consistent with the 5% weekly turnover mentioned previously. This is because the same blocks are changed frequently across the week, even though a single database might only have a 5% change rate per week based on the database. Regular backups capture these incremental changes.

Policies

Two options exist for tiering inactive snapshot blocks to the capacity tier. First, the `snapshot-only` policy only targets the snapshot blocks. Although the `auto` policy includes the `snapshot-only` blocks, it also tiers blocks from the active file system. This might not be desirable.

The `tiering-minimum-cooling-days` value should be set to a time period that makes data that might be required during a restoration available on the performance tier. For example, most restore scenarios of a critical production database include a restore point at some time in the previous few days. Setting a `tiering-minimum-cooling-days` value of `3` makes sure that any restoration of the database results in a database that immediately performs at its best. All blocks in the active data files are still present on fast storage without needing to recover them from the capacity tier.

External backups

You should use NetApp Snapshot™ copies as your primary method of database data protection because doing so provides space-efficient and nearly instantaneous backup and recovery capability irrespective of the size of the database. Snapshot copies should not, however, be the only option for recovery because they share space on the same drives as the active data. If the storage system is destroyed, then the backups are also destroyed.

Although nearly all restore scenarios can be performed by using local Snapshot copies, any important database requires a copy on different media. The two most common options are replicated snapshots or a traditional file-based backup.

Note: You should make sure that the capacity tier serving replicated snapshot data is independent of the primary data location to avoid creating a single point of failure. The purpose of an external backup is to make sure that a single storage system failure does not affect both the primary and replicated copies of your data.

Snapshot replication

A Snapshot copy replicated with SnapMirror or SnapVault that is only used for recovery should generally use the FabricPool `all` policy. With this policy, metadata is replicated, but all data blocks are immediately sent to the capacity tier, which yields maximum performance. Most recovery processes involve sequential I/O, which is inherently efficient. The recovery time from the object store destination should be evaluated, but, in a well-designed architecture, this recovery process does not need to be significantly slower than recovery from local data.

If the replicated data is also intended to be used for cloning, the `auto` policy is more appropriate, with a `tiering-minimum-cooling-days` value that encompasses data that is expected to be regularly used in a cloning environment. For example, a database's active working set might include data read or written in the previous three days, but it could also include another six months of historical data. If so, then the `auto` policy at the SnapMirror destination makes the working set available on the performance tier.

Traditional backups

Traditional backups include products such as Oracle Recovery Manager, which create file-based backups outside the location of the original database.

A `tiering-minimum-cooling-days` policy of a few days preserves the most recent backups, and therefore the backups most likely to be required for an urgent recovery situation, on the performance tier. The data blocks of the older files are then moved to the capacity tier.

The `auto` policy is the most appropriate policy for backup data. This ensures prompt tiering when the cooling threshold has been reached irrespective of whether the files have been deleted or continue to exist in the primary file system. Storing all the potentially required files in a single location in the active file system also simplifies management. There is no reason to search through snapshots to locate a file that needs to be restored.

The `snapshot-only` policy can be made to work, but that policy only applies to blocks that are no longer in the active file system. Therefore, you must delete archived logs on an NFS or SMB share first before the data can be tiered.

Tiering is even less efficient with a LUN configuration because deletion of a file from a LUN only removes the file references from the file system metadata. The actual blocks on the LUNs remain in place until they are overwritten. This situation can create a lengthy delay between the time a file is deleted and the time that the blocks are overwritten and become candidates for tiering. There is some benefit to moving the `snapshot-only` blocks to the capacity tier, but, overall, FabricPool management of backup data works best with the `auto` policy.

Object store access interruptions

If an I/O issued to ONTAP requires data from the capacity tier and ONTAP cannot reach the capacity tier to retrieve blocks, then the I/O eventually times out. The effect of this timeout depends on the protocol used. In an NFS environment, ONTAP responds with either an EJUKEBOX or EDELAY response, depending on the protocol. Some older operating systems might interpret this as an error, but current operating systems and current patch levels of the Oracle Direct NFS client treat this as a retriable error and continue waiting for the I/O to complete.

A shorter timeout applies to SAN environments. If a block in the object store environment is required and remains unreachable for two minutes, a read error is returned to the host. The ONTAP volume and LUNs remain online, but the host operating system might flag the file system as being in an error state.

Object storage connectivity problems with the `snapshot-only` policy is less of a concern because only backup data is tiered. Communication problems can slow data recovery but not otherwise affect data being actively used. The `auto` and `all` policies allow tiering of cold data from the **active** LUN, which means that an error during object store data retrieval can affect database availability. You should only use a SAN deployment with these policies with enterprise-class object storage and network connections designed for high availability. NetApp StorageGRID is the superior option.

Conclusion

NetApp FabricPool is an optimal solution for databases to improve efficiency and manageability. You can make sure that your high-performance storage system is hosting high-performing critical data while relocating the colder data to an object store system, including private and public cloud options. Manageability is unimpaired because the metadata is centralized, and you can see all your data as usual. ONTAP FabricPool automatically manages the location of the underlying data blocks.

Where to find additional information

To learn more about the information described in this document, refer to the following documents and/or websites:

- TR-4598: FabricPool Best Practices
  http://www.netapp.com/us/media/tr-4598.pdf
- TR-3633: Oracle Databases on ONTAP
  http://www.netapp.com/us/media/tr-3633.pdf
- TR-4591: Database Data Protection
  http://www.netapp.com/us/media/tr-4591.pdf
- TR-4592: Oracle on MetroCluster
  http://www.netapp.com/us/media/tr-4592.pdf
- TR-4534: Migration of Oracle Databases to NetApp Storage Systems
  http://www.netapp.com/us/media/tr-4534.pdf

Version history

Version	Date	Document version history
Version 1.0	April 2018	Initial release
Version 1.2	April 2021	Update for ONTAP 9.8

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.

Copyright Information

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

Data contained herein pertains to a commercial item (as defined in FAR 2.101) and is proprietary to NetApp, Inc. The U.S. Government has a non-exclusive, non-transferrable, non-sublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b).

Trademark Information