Technical Report

# Database Storage Tiering with NetApp FabricPool

Jeffrey Steiner, BS Navyashree, Srinivas Venkat, NetApp

June 2018 | 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).

NetApp®

TABLE OF CONTENTS


1 Introduction ..... 3

1.1 FabricPool and NVMe ..... 3

2 FabricPool Overview ..... 3

2.1 Architecture ..... 3

2.2 Support ..... 4

2.3 Object Store Providers ..... 4

2.4 Data and Metadata ..... 5

2.5 Backups ..... 5

2.6 Tiering Policies ..... 5

3 Databases with FabricPool Designs ..... 6

3.1 FabricPool and Database Workloads ..... 6

3.2 Log Archiving ..... 8

3.3 Full Datafile Tiering ..... 9

3.4 Database Block Tiering ..... 9

3.5 Local Snapshot-Based Backups ..... 10

3.6 External Backups ..... 10

3.7 Object Store Access Interruptions ..... 11

Conclusion ..... 12

Where to Find Additional Information ..... 12

# 1 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.

## 1.1 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.

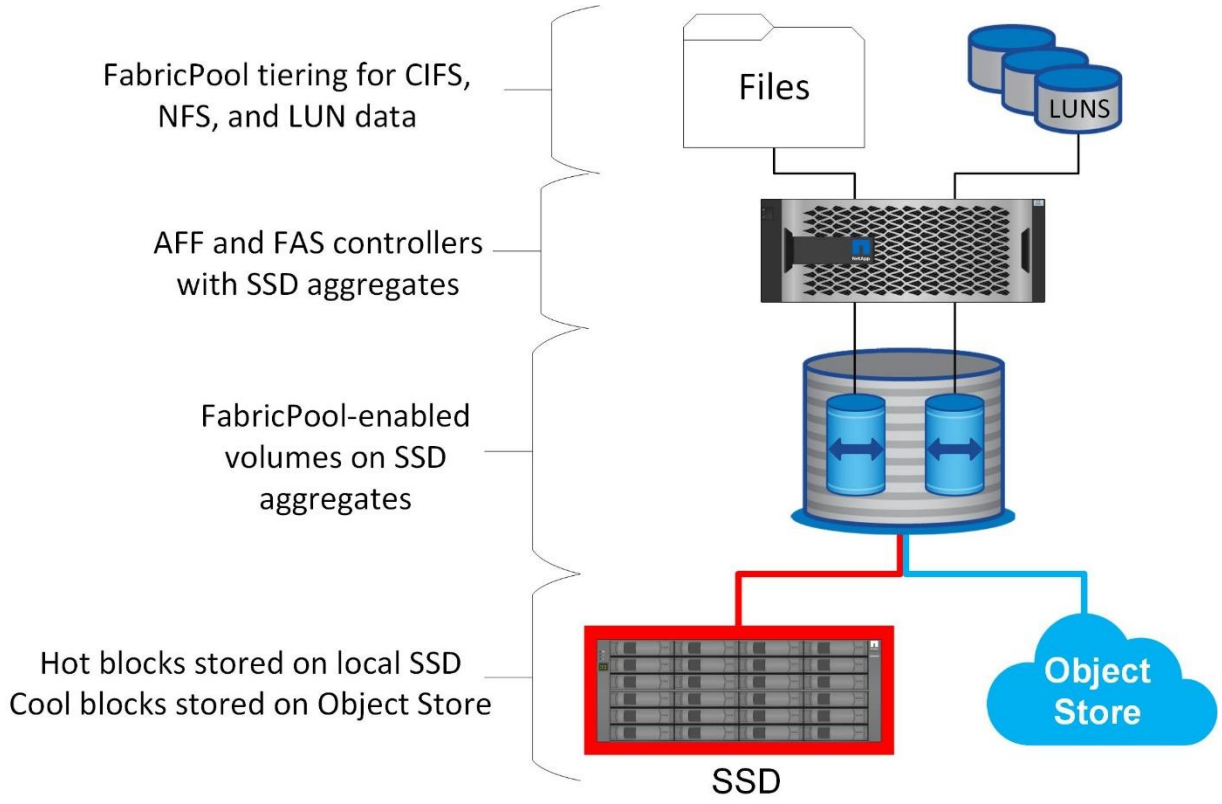
# 2 FabricPool Overview

## 2.1 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®, Microsoft Azure Blob storage, and Amazon AWS 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 filesystem 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 appropriate tier.

**Figure 1) FabricPool architecture.**




## 2.2 Support

Support is included for the following configurations:

- AFF systems
- SSD aggregates on FAS systems
- ONTAP Select. NetApp recommends using all SSD FabricPool aggregates.
- ONTAP Cloud with gp2 and st1 Amazon Elastic Block Store (EBS) volumes

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

## 2.4 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.

## 2.5 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.

## 2.6 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 9.4. There are two aspects to tiering behavior:

- How does a block on the performance tier become a candidate to be relocated to the capacity tier?
- What happens when a block currently on the capacity tier is read by the application?

### Snapshot-Only

The Snapshot-Only policy applies only to blocks that are not shared with the active filesystem. 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. The range is from 2 to 63 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 filesystem and thus would not be candidates for tiering under this policy.

When a block currently on the capacity tier is read, ONTAP registers that data is being read with the intent to make it active again. Data is therefore returned to the performance tier. This approach addresses database recovery scenarios. If data within a snapshot is tiered out, the only expected reason the data would be read again would be as part of a database restoration. Therefore, the read process should make the data blocks hot again.

### Auto

The Auto tiering policy extends tiering to both snapshot-specific blocks as well as blocks within the active filesystem. The delay before a block is considered cool is controlled by the `tiering-minimum-cooling-days` setting for the volume. The range is from 2 to 63 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 3 days results in any log files older than 3 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 3.2, “Log Archiving.”

The effect of reading cooled blocks depends on the type of I/O being performed. A random read of a cooled block under the Auto policy causes the block to be made hot again. The presumption is that the block is being read for a reason and is needed again. Cooled data that is read sequentially is not made hot again, because it is assumed that the data is being used for a recovery operation and is only required one time.

### 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 could become useful to change the policies over time. For example, let’s say that a specific database is extensively tiered to the capacity layer, but an unexpected need for full performance capabilities arises. The policy can be changed to prevent any additional tiering and to confirm that any blocks read back as the database becomes more active remain in the performance tier.

### Backup

The Backup policy applies only to data protection volumes, meaning a SnapMirror or NetApp SnapVault® destination. All blocks that are transferred to the destination are immediately considered cool and eligible to be tiered to the capacity layer. This approach is especially appropriate for extremely long-term backups.

**Note:** If the replication relationship is broken, the policy automatically changes to Auto.

# 3 Databases with FabricPool Designs

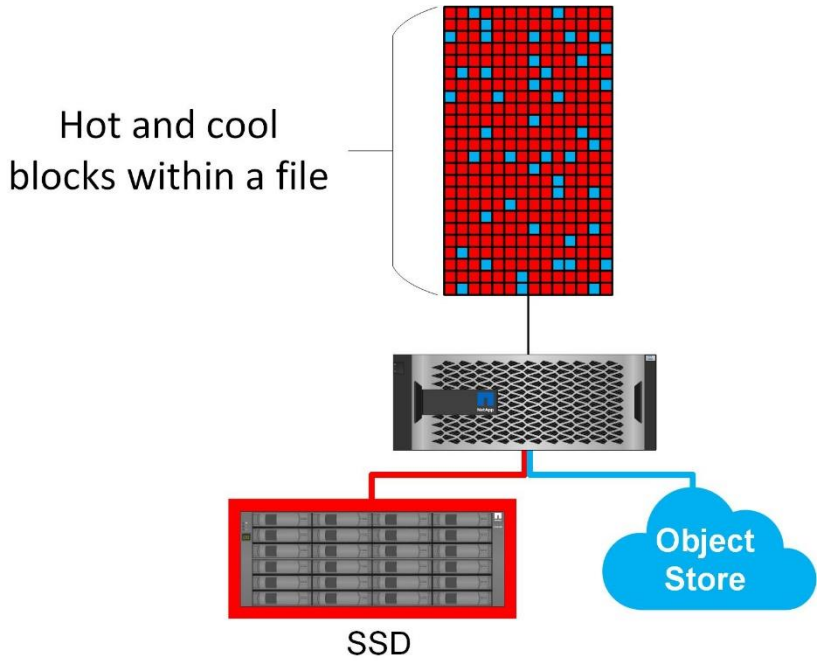
There are multiple ways to use FabricPool with database workloads and datasets. The following sections explain the basic options and are not mutually exclusive.

## 3.1 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 datafile 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.

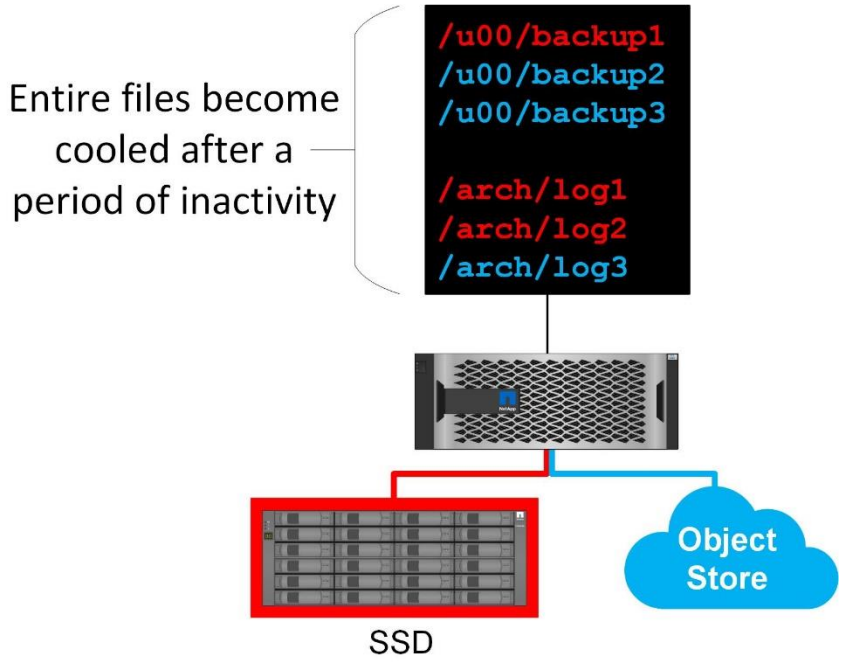
**Figure 2) Sub-file block tiering.**




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.

## 3.2 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 datafiles 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 filesystem. This is not only complicated, but also risky because a tape-based backup system is much more prone to problems that interrupt service.

Fabric Pools 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 the most appropriate policy for a volume hosting archive log data. This 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 filesystem. Storing all the potentially required logs in a single location in the active filesystem 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 filesystem. Therefore, archived logs on an NFS or SMB share would need to be deleted before the data could be tiered. Tiering would be even less efficiency with a LUN configuration because deletion of a file from a LUN only removes the file references from the filesystem metadata. The actual blocks on the LUNs remain in place until overwritten. This could 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 are some benefit to moving the Snapshot-Only blocks to the capacity tier, but overall FabricPool management of archive log data works best with the Auto policy.

A SnapMirror or SnapVault destination volume containing archived log files can also use a FabricPool Backup 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.

## 3.3 Full Datafile Tiering

Many databases include datafiles 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. FabricPool can be used to relocate older datafiles to the capacity tier.

Some databases, such as an Oracle database using the Information Lifecycle Management assistant, can relocate datafiles between filesystems based on certain criteria. Such a system could be used 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 could 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` threshhold 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 would 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 datafiles because only the Auto policy affects blocks that are in the active filesystem.

**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` threshhold is never reached.

## 3.4 Database Block Tiering

Datafiles 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, datafiles 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 2 days would tier the data promptly after the data becomes less active.

The Auto policy is required to drive tiering of datafile blocks because only the Auto policy affects blocks that are in the active filesystem.

## 3.5 Local Snapshot-Based Backups

The initial release of FabricPool targeted the backup use case. The only type of blocks that could be tiered were blocks that were no longer associated with data in the active filesystem. Therefore, only the snapshot data blocks could be moved 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 datafiles 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 filesystem. 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 would make sure that any restoration of the database results in a database that immediately performs at its best. All blocks in the active datafiles are still present on fast storage without needing to recover them from the capacity tier.

## 3.6 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 Backup 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 6 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 filesystem. Storing all the potentially required files in a single location in the active filesystem also simplifies management. There is no reason to search through snapshots to locate a file that needs to be restored.

The Snapshot-Only policy could be made to work, but that policy only applies to blocks that are no longer in the active filesystem. Therefore, archived logs on an NFS or SMB share must be deleted 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 filesystem 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.

## 3.7 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.

As of the time of writing, a shorter timeout applies to SAN environments. If a block in the object store environment is required and remains unreachable for 10 seconds, a read error is returned to the host. The ONTAP volume and LUNs remain online, but the host OS might flag the filesystem as being in an error state. With a FabricPool policy of Backup or Snapshot-Only, there is a delay in retrieving data from backups.

However, FabricPool with the Auto policy should be used with enterprise-class object storage because the Auto policy allows tiering of cold data from the active filesystem. If an error occurs when retrieving

data, access to the LUN hosting both the hot and cold data might be disrupted, affecting overall database availability. A SAN deployment with the FabricPool Auto policy should only be used with enterprise-class object storage and network connections designed for high availability. NetApp StorageGRID Webscale 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

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.



TR-4676-0418