



技术报告

使用 **NetApp** 进行数据库存储分层

NetApp公司Srinivas Venkat公司BS Navyashree Jeffrey Steiner

2021年4月 | TR-4695

摘要

本文档介绍了**NetApp® FabricPool®**在各种数据库中的优势和配置选项、其中包括**Oracle**关系数据库管理系统(RDBMS)。

目录

引言	3
FabricPool 和NVMe	3
FabricPool 概述	3
架构	3
支持	4
对象存储提供程序	4
数据和元数据	5
备份	5
分层策略	5
检索策略	6
采用FabricPool 设计的数据库	7
FabricPool 和数据库工作负载	7
日志归档	9
完整数据文件分层	10
数据库块分层	10
基于快照的备份	11
外部备份	11
对象存储访问中断	12
结束语	13
版本历史记录	13

插图目录

图 1) FabricPool 架构	4
图 2) 子文件块分层	8
图 3) 完整文件块分层	8

简介

NetApp FabricPool 是一种自动化存储分层功能、其中活动数据驻留在本地高性能固态驱动器(SSD)上、而非活动数据分层到低成本对象存储。它最初在NetApp ONTAP® 9.2中推出、用于管理只读数据、之后在ONTAP 9.4中进行了增强、还对活动数据进行了分层。

在数据库环境中、您可以创建一个存储架构、其中热数据会保留在本地存储阵列上。归档日志、数据库备份等非活动数据、甚至非活动数据库块也会移至成本较低的对象存储。

FabricPool 与ONTAP 集成在一起。分层过程通过多个基于策略的管理选项实现了完全自动化。除了对象存储层的性能特征之外、FabricPool 对应用程序和数据库配置是透明的。无需更改架构、您可以继续从中央ONTAP 存储系统管理数据库和应用程序环境。

FabricPool 和NVMe

您还应将FabricPool 视为非易失性内存快速(NVMe)策略的重要组成部分。FabricPool 将NVMe性能层与对象存储容量层集成在一起。只有真正属于基于NVMe的介质的数据才会保留在性能层上、从而为您的NVMe投资提供最大回报。

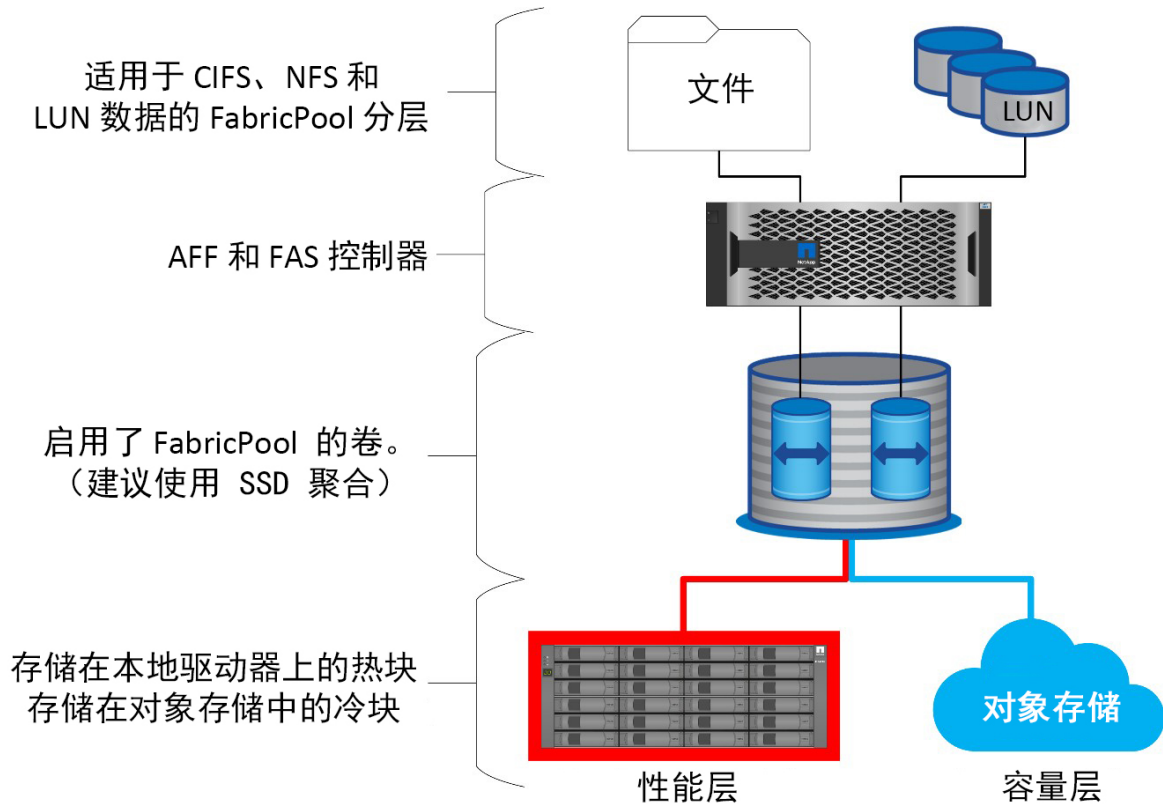
FabricPool 概述

架构

FabricPool 是一种分层技术、可将块分为热块或冷块、并将其放置在最合适的存储层中。性能层位于SSD存储上、并托管热数据块。容量层位于对象存储目标上、并托管冷数据块。对象存储支持包括NetApp StorageGRID®、NetApp ONTAP S3、Microsoft Azure Blob存储、Alibaba云对象存储、IBM云对象存储、Google Cloud和Amazon Simple Storage Service (Amazon S3)。图 1 显示了 FabricPool 架构。

可以使用多个分层策略来控制块的分类方式、这些策略可以按卷进行设置、并根据需要进行更改。在性能层和容量层之间仅移动数据块。定义LUN和文件系统结构的元数据始终保留在性能层上。因此、管理集中在ONTAP 上。文件和LUN与任何其他ONTAP 配置上存储的数据没有任何不同。NetApp AFF 或FAS 控制器会应用定义的策略将数据移动到合适的Jefopriate层。

图 1) FabricPool 架构。



支持

支持以下配置：

- AFF 系统
- 大多数支持运行ONTAP 9.2的FAS 系统。建议使用SSD聚合、但也支持HDD。
- ONTAP SelectNetApp建议使用所有SSD FabricPool 聚合。
- ONTAP Cloud

对象存储提供程序

对象存储协议使用简单、HTTP或HTTPS请求来存储大量数据对象。对对象存储的访问必须可靠、因为从ONTAP 访问数据取决于请求的及时服务。选项包括Amazon S3 Standard和Infrequent Access选项以及Microsoft Azure热和冷Blob Storage、IBM Cloud和Google Cloud。不支持Amazon Glacier和Amazon Archive等归档选项、因为检索数据所需的时间可能超过主机操作系统和应用程序的容差。

NetApp StorageGRID 也受支持、是最佳的企业级解决方案。它是一个高性能、可扩展且高度安全的对象存储系统、可以为FabricPool 数据以及越来越可能属于企业数据库环境的其他对象存储应用程序提供地理冗余。

StorageGRID 还可以避免许多公有云提供商因从其服务中读取数据而收取的出站费用、从而降低成本。

数据和元数据

请注意、此处的术语"数据"适用场景是指实际数据块、而不是元数据。仅对数据块进行分层、而元数据仍保留在SSD的本地。此外、只有读取实际数据块、数据块的热或冷状态才会受到影响。只读文件的名称、时间戳或所有权元数据不会影响底层数据块的位置。

备份

虽然FabricPool 可以显著减少存储占用空间、但它本身并不是备份解决方案。NetApp WAFL® 元数据始终位于性能层上。如果灾难性灾难破坏了性能层、则无法使用容量层上的数据创建新环境、因为它不包含WAFL 元数据。

但是、FabricPool 可以成为备份策略的一部分。例如、可以为FabricPool 配置NetApp SnapMirror® 复制技术。镜像的每一半都可以与对象存储目标建立自己的连接。结果是两个独立的数据副本。主副本由性能层上的块和容量层中的关联块组成、副本是第二组性能和容量块。

分层策略

NetApp FlexVol® 卷是ONTAP 的基本管理单元。FlexVol 卷本身不会占用空间、而是只是文件和/或LUN的容器。然后根据需要在卷上设置策略、包括FabricPool 分层策略。

ONTAP 中提供了四个策略、用于控制性能层上的块如何成为要重新定位到容量层的候选块。

仅快照

snapshot-only tiering-policy 仅适用于未与活动文件系统共享的块。从本质上讲、它会对数据库备份进行分层。创建快照并覆盖块后、块将成为分层的候选块、从而导致只有该快照副本中存在一个块。snapshot-only 块被视为冷却之前的延迟由 tiering-minimum-cooling-days 卷的设置控制。从ONTAP 9.8及更高版本开始、此范围为2到183天。

大多数数据库的更改率较低、因此此策略节省的空间极少。例如、在ONTAP 上观察到的典型数据库的变更率每周低于5%。存在例外情况、但速率通常较低。数据库归档日志可能会占用大量空间、但它们通常仍存在于活动文件系统中、因此不适合在此策略下分层。

自动

auto 分层策略可将分层扩展到特定于快照的块以及活动文件系统中的块。块被视为冷却之前的延迟由 tiering-minimum-cooling-days 卷的设置控制。从ONTAP 9.8及更高版本开始、此范围为2到183天。

此方法可启用 snapshot-only 策略不可用的分层选项。例如、数据保护策略可能需要保留90天的归档日志文件。如果将散热期设置为三天、则会导致超过三天的任何日志文件从性能层分层出来。此操作可释放高速存储平台上的大量空间、同时仍允许您查看和管理90天的完整数据。有关详细信息、请参见第 0 节"日志归档"。"

无

`none` 此策略可防止从存储层分层任何其他块、但容量层中仍存在的任何数据仍会保留在容量层中、直到读取为止。如果随后读取该块、则会将其拉回并放置在性能层上。

使用 `none` 此策略的主要原因是防止对块进行分层、但它对于随着时间的推移更改策略非常有用、例如、如果某个特定数据库已广泛分层到容量层、但出现了完整性功能的意外需求。您可以更改此策略、以防止发生任何其他分层、并确认在数据库变得更活跃时读取回的任何块仍保留在性能层中。

全部

`all` 从 `backup ONTAP 9.6` 开始、此策略将取代此策略。`backup` 此策略仅应用于数据保护卷、即 `SnapMirror` 或 `NetApp SnapVault®` 目标。`all` 此策略的工作方式相同、但不限于数据保护卷。

使用此策略时、块会立即视为冷块、并可立即分层到容量层。

此策略尤其适用于长期备份。您可以将其用作分层存储管理(HSM)的一种形式。过去、通常使用HSM将文件的数据块分层到磁带、同时保持文件本身在文件系统上可见。使用 `all` 具有策略的 `FabricPool` 卷可以以可查看且可管理的方式存储文件、但几乎不会占用本地存储层上的任何空间。

检索策略

分层策略控制从性能层到容量层的分层块。检索策略用于控制读取已分层块时发生的情况。

默认

所有 `FabricPool` 卷最初都设置为 `default`、这意味着行为由控制 `cloud-retrieval-policy`、具体行为取决于所使用的分层策略。

- `auto`。仅检索随机读取的数据。
- `snapshot-only`。检索所有按顺序或随机读取的数据。
- `none`。检索所有按顺序或随机读取的数据。
- `all`。不从容量层检索数据。

读写

如果设置 `cloud-retrieval-policy` 为"读"、则会覆盖默认行为、以便读取任何分层数据后、数据将返回到性能层。

例如 `auto`、在分层策略下、卷可能已长时间被少量使用、并且大多数块现在已分层。如果业务需求发生意外变化、需要重复扫描某些数据以准备特定报告、则可能需要将 `cloud-retrieval-policy` 更改为 `on-read`、以确保读取的所有数据都返回到性能层、包括顺序读取和随机读取的数据。这样可以提高卷的顺序I/O操作的性能。

提升

提升策略的行为取决于分层策略。如果分层策略为 `auto`、则将设置 `cloud-retrieval-policy` 为 `promote` 在下次分层扫描时从容量层返回所有块。

如果分层策略为 `snapshot-only`、则返回的唯一块是与活动文件系统关联的块。通常、这不会产生任何影响、因为 `snapshot-only` 根据策略分层的唯一块是专用于快照的块。活动文件系统中没有分层块。但是、如果通过卷 `SnapRestore` 或文件克隆操作从快照还原卷上的数据、则活动文件系统可能会需要一些由于仅与快照关联而分层出来的块。可能需要暂时将 `cloud-retrieval-policy` 策略更改为 `promote`、以便快速检索所有本地所需的块。

从不

请勿从容量层检索块。

采用FabricPool 设计的数据库

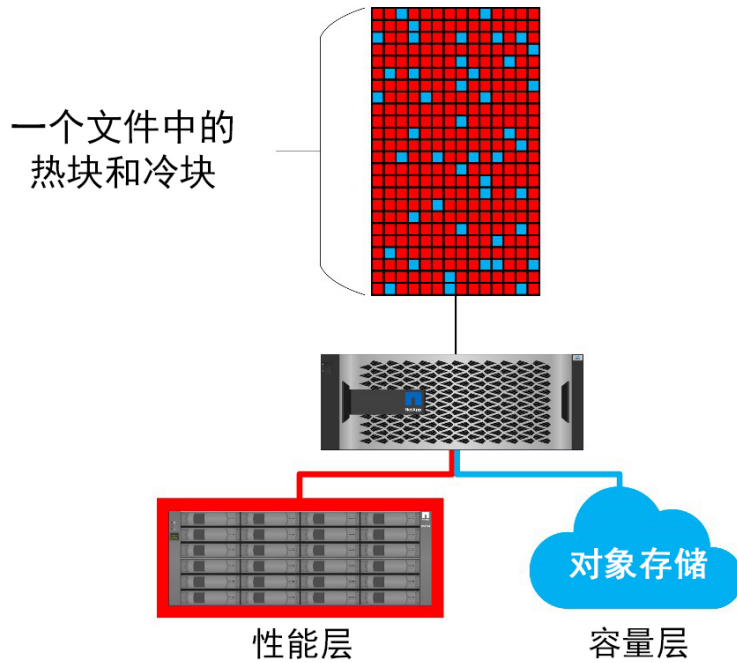
可以通过多种方法将 `FabricPool` 用于数据库工作负载和数据集。以下各节介绍了基本选项、它们并不是互斥的。

FabricPool 和数据库工作负载

从 `ONTAP` 的角度来看、`FabricPool` 在块级别运行、但有效结果使您能够对整个文件以及较大文件中的块进行分层。

例如、数据库数据文件上的主 `I/O` 模式是随机读写。采用 `auto` 此策略的 `FabricPool` 可提供一个卷、其中包含 `SSD` 性能层上的活动块、而冷块则重新定位到容量层。数据库本身仍可在一个位置看到普通文件、但数据块的实际位置如 图2 所示。

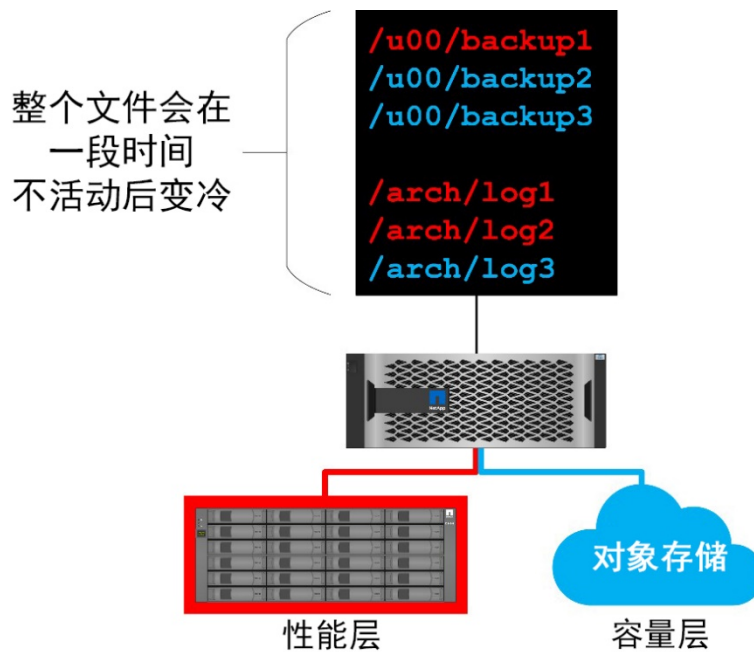
图2)子文件块分层。



在其他情况下、整个文件将变为非活动状态。其中包括归档事务日志或平面文件备份等文件。这些类型的文件通常会联机若干天、以便在需要数据库恢复时提供高速访问。但是、几天后、很可能再次需要这些资源。这些类型的文件通常写入一次、之后不会访问。

请务必了解、FabricPool 在块级别运行、而不是在文件级别运行。如果数据集包含不再访问的文件、则这些整个文件将分层到容量层(图3)。

图3)整个文件块分层。



注意： 任何类型的数据访问都会重置热图数据。因此、数据库全表扫描甚至读取源文件的备份活动都无法分层、因为 `tiering- minimum-cooling-days` 永远不会达到所需阈值。

日志归档

FabricPool 最重要的用途可能是提高归档日志文件管理的效率。大多数关系数据库都在日志归档模式下运行、以提供时间点恢复。通过在事务日志中记录更改来提交对数据库的更改、事务日志会保留下来而不被覆盖。在某些情况下、事务日志会作为活动日志文件的副本创建。在其他情况下、活动日志文件将关闭、数据库将在同一位置使用不同名称创建新的日志文件。

要恢复数据库、可以将数据文件还原到所需恢复点之前的某个时间、然后重放事务日志、直到数据库处于所需状态为止。

因此、可能需要保留大量已归档的事务日志。所需容量可以是数据库本身大小的许多倍。**NetApp**发现数据库中有超过**100 TB**的归档日志、这些日志保持联机并可用于满足法规要求。

这种情况会给管理带来巨大挑战。事务日志可以保留在磁盘上、但这会造成成本高昂和浪费、尤其是在数据库迁移到基于**NVMe**的存储时。可以专门为日志创建不同的**SATA**存储层、但这种方法会创建一个额外的存储类型、必须进行部署和管理。还有一些客户继续使用基于磁带的备份解决方案、将日志扫描到磁带并将其从活动文件系统中删除。这不仅复杂、而且风险很大、因为基于磁带的备份系统更容易出现中断服务的问题。

FabricPool 通过提供一个具有成分层功能的解决方案 来解决这些问题。事务日志会进行存储、并可在其正常位置访问。

策略

`tiering-minimum-cooling-days` 几天的策略会在性能层上保留最新的日志、因此在紧急恢复情况下最可能需要的日志。然后、旧文件中的数据块将移至容量层。

`auto` 对于托管**活动文件系统**中的归档日志数据的卷、此策略通常是最合适的策略。**Oracle**等适用场景 数据库对同一组重做日志文件执行循环覆盖。

`auto` 无论日志是否已删除或继续存在于主文件系统中、此策略都会在达到冷却阈值时强制执行提示分层。将所有可能需要的日志存储在活动文件系统的同一个位置也可以简化管理。没有理由通过快照搜索来查找需要还原的文件。

某些数据库(例如**Microsoft SQL Server**)会在备份操作期间截断事务日志文件、以使日志不再位于活动文件系统中。您可以使用 `snapshot- only` 分层策略节省容量、但 `auto` 此策略对于日志数据不有用、因为活动文件系统中很少有冷却的日志数据。

`snapshot-only` 该策略仅对不再位于活动文件系统中的块进行分层。因此、您必须先删除**NFS**或**SMB**共享上的归档日志、然后才能对数据进行分层。

对于**LUN**配置、分层效率甚至会降低、因为从**LUN**中删除文件仅会从文件系统元数据中删除文件引用。**LUN**上的实际块将一直保留在原位、直到被覆盖为止。这可能会造成从删除文件到覆盖块并成为分层候选块之间的长时间延迟。

包含归档日志文件 `all` 的 `SnapMirror` 或 `SnapVault` 目标卷也可以使用 `FabricPool` 策略立即对所有数据进行分层、但节省的空间可能并不多。 `auto` 此策略通常会在性能层留下非常少的数据。如果出于恢复目的需要这些副本、最好在尽可能快的层上提供最近创建的文件。

完整数据文件分层

许多数据库都包含按日期组织的数据文件、随着时间的老化、此类数据的访问可能性通常会越来越小。例如、大型计费数据库可能包含五年的客户数据、但只有最近几个月处于活动状态。您可以使用 `FabricPool` 将旧数据文件重新定位到容量层。

某些数据库(例如使用信息生命周期管理助理的 `Oracle` 数据库)可以根据特定标准在文件系统之间重新定位数据文件。您可以将此类系统与 `FabricPool` 结合使用、将文件从活动数据卷重新定位到归档数据卷。例如、包含最近三个月数据的主数据文件可能位于没有 `FabricPool` 的 `SSD` 存储上。90天后、各个数据文件将移动到配置有两天冷却期的卷、以推动快速分层到容量层。

此外、数据文件还可以共享启用了 `FabricPool` 的通用卷、并且冷却期更长、以确保所需数据在性能层上保持可用。数据可能会比预期更早地重新定位到容量层、但只有在所需时间段内未访问数据后、才会应用冷却期。数据不会仅仅因为冷却期已过而立即分层。

注意： 任何类型的数据访问都会重置热图数据。因此、数据库完整表扫描甚至读取源文件的备份活动都无法分层、因为 `tiering- minimum-cooling-days` 永远不会达到所需阈值。

策略

`tiering- minimum-cooling-days` 此策略应设置得足够高、以便您可能需要的文件仍保留在性能层上。例如、需要最新60天数据且性能最佳的数据库需要 `tiering- minimum-cooling-days` 将时间段设置为 60。根据文件访问模式、也可以获得类似的结果。例如、如果需要最近90天的数据、而应用程序正在访问该90天的数据、则这些数据将保留在性能层上。通过将 `tiering- minimum-cooling-days` 期限设置为 2、您可以在数据不活跃时进行提示分层。

`auto` 需要使用此策略来对数据文件进行分层、因为只有 `auto` 此策略会影响活动文件系统中的块。

注意： 任何类型的数据访问都会重置热图数据。因此、数据库完整表扫描甚至读取源文件的备份活动都无法分层、因为 `tiering- minimum-cooling-days` 永远不会达到所需阈值。

数据库块分层

已知包含非活动块的数据文件也可进行 `FabricPool` 分层。例如、供应链管理数据库可能包含一些历史信息、这些信息在需要时必须可用、但在正常操作期间不会访问。可以使用 `FabricPool` 有选择地重新定位非活动块。

例如、在具有的 `FabricPool` 卷上运行的数据文件 `tiering- minimum-cooling-days` 在90天期限内、会保留在性能层上过去90天内访问的任何块。但是、如果在90天内未访问任何内容、则会将其重新定位到容量层。在其他情况下、正常的应用程序活动会将正确的块保留在正确的层上。例如、如果通常使用数据库定期处理前60天的数据、 `tiering- minimum-cooling-days` 则可以设置更短的时间段、因为应用程序的自然活动可以确保块不会过早地重新定位。

注意： 任何类型的数据访问都会重置热图数据。因此、数据库完整表扫描甚至读取源文件的备份活动都无法分层、因为 `tiering- minimum-cooling-days` 永远不会达到所需阈值。

策略

`tiering-minimum-cooling-days` 策略应设置得足够高、以保留性能层上可能需要的文件。例如、如果数据库可能需要最新60天的数据并获得最佳性能 `tiering-minimum-cooling-days`、则需要将时间段设置为60天。根据文件的访问模式、也可以获得类似的结果。例如、如果需要最近90天的数据、而应用程序正在访问该90天的数据、则这些数据将保留在性能层上。如果将 `tiering-minimum-cooling-days` 时间段设置为两天、则会在数据不活跃后立即对数据进行分层。

`auto` 要对数据文件块进行分层、需要使用此策略、因为只有 `auto` 此策略会影响活动文件系统中的块。

基于Snapshot的本地备份

FabricPool 的初始版本针对备份用例。您可以分层的唯一块类型是不再与活动文件系统中的数据关联的块。因此、您只能将快照数据块移动到容量层。

在NetApp存储上观察到的典型数据库每周的营业额约为5%。但一般而言、营业额较低。一种典型的基于Snapshot的备份方法是每6到24小时备份一次数据文件。因此、每周的总快照空间消耗约为总数据库大小的15%。起初、此百分比似乎与前面提到的5%的每周营业额不一致。这是因为即使一个数据库每周的更改率可能只有5%、同一个块也会在一周内频繁更改。定期备份会捕获这些增量更改。

策略

可以通过两种方法将非活动快照块分层到容量层。首先 `snapshot-only`、该策略仅针对快照块。`auto` 尽管此策略包含 `snapshot-only` 块、但它也会对活动文件系统中的块进行分层。这可能并不可取。

`tiering-minimum-cooling-days` 此值应设置为一个时间段、以便在性能层上提供还原期间可能需要的数据。例如、关键生产数据库的大多数还原情形都包括前几天某个时间的还原点。如果 `tiering-minimum-cooling-days` 将值设置为 3、则可以确保对数据库进行任何还原都能使数据库立即达到最佳性能。活动数据文件中的所有块仍位于快速存储中、无需从容量层恢复。

外部备份

您应使用NetApp Snapshot® 副本作为数据库数据保护的主要方法、因为无论数据库大小如何、这样做都可以节省空间并近乎即时地进行备份和恢复。但是、Snapshot副本不应是唯一的恢复选项、因为它们与活动数据在相同的驱动器上共享空间。如果存储系统被销毁、则备份也会被销毁。

尽管几乎所有还原方案都可以使用本地Snapshot副本执行、但任何重要数据库都需要在不同介质上进行副本。两个最常见的选项是复制快照或基于文件的传统备份。

注意： 您应确保提供复制快照数据的容量层独立于主数据位置、以避免创建单点故障。外部备份的目的是确保单个存储系统故障不会同时影响数据的主副本和复制副本。

快照复制

使用SnapMirror或SnapVault 复制且仅用于恢复 all 的Snapshot副本通常应使用FabricPool 策略。使用此策略、元数据会进行复制、但所有数据块会立即发送到容量层、从而实现最高性能。大多数恢复过程都涉及顺序I/O、而顺序I/O本身就很高效率。应评估从对象存储目标恢复的时间、但在设计完善的架构中、此恢复过程不需要比从本地数据恢复要慢得多。

如果复制的数据也要用于克隆、auto 则策略更为合适、tiering-minimum-cooling-days 其值应包含克隆环境中应定期使用的数据。例如、数据库的活动工作集可能包括前三天读取或写入的数据、但也可能包括另外六个月的历史数据。如果是、auto 则SnapMirror目标上的策略会使工作集在性能层上可用。

传统备份

传统备份包括Oracle Recovery Manager等产品、这些产品可在原始数据库所在位置以外创建基于文件的备份。

tiering-minimum-cooling-days 几天的策略会在性能层保留最新的备份、因此紧急恢复情况下最可能需要的备份。然后、旧文件的数据块将移至容量层。

auto 此策略是最适合备份数据的策略。这样可以确保在达到散热阈值时、无论文件是否已删除或是否继续存在于主文件系统中、都能进行提示分层。将所有可能需要的文件存储在活动文件系统的同一个位置也可以简化管理。没有理由通过快照搜索来查找需要还原的文件。

snapshot-only 可以使此策略生效、但该策略仅限不再位于活动文件系统中的适用场景 块。因此、必须先删除NFS或SMB共享上的归档日志、然后才能对数据进行分层。

对于LUN配置、分层效率甚至会降低、因为从LUN中删除文件仅会从文件系统元数据中删除文件引用。LUN上的实际块将一直保留在原位、直到被覆盖为止。这种情况可能会造成从删除文件到覆盖块并成为分层候选块之间的长时间延迟。将 snapshot-only 块移至容量层有一定的优势、但总体而言、FabricPool 备份数据管理与 auto 策略最适合。

对象存储访问中断

如果向ONTAP 发出的I/O需要容量层中的数据、而ONTAP 无法到达容量层来检索块、则I/O最终会超时。此超时的影响取决于所使用的协议。在NFS环境中、ONTAP 会根据协议使用EJUKEBOX或EDELTY响应进行响应。某些较旧的操作系统可能会将此错误解释为错误、但Oracle Direct NFS客户端的当前操作系统和当前修补程序级别会将此错误视为可检索的错误、并继续等待I/O完成。

适用场景 SAN环境的超时时间较短。如果需要对象存储环境中的某个块、并且该块在两分钟内无法访问、则会向主机返回读取错误。ONTAP 卷和LUN保持联机状态、但主机操作系统可能会将文件系统标记为处于错误状态。

snapshot-only 此策略中的对象存储连接问题不值得关注、因为只会对备份数据进行分层。通信问题可能会减慢数据恢复速度、但不会影响当前正在使用的数据。auto 和 all 策略允许对 **活动 LUN**中的冷数据进行分层、这意味着对象存储数据检索期间出错可能会影响数据库可用性。您只能将SAN部署与这些策略结合使用、并使用专为实现高可用性而设计的企业级对象存储和网络连接。NetApp StorageGRID 是最佳选择。

结论

NetApp FabricPool 是适用于数据库的最佳解决方案、可提高效率和易管理性。您可以确保高性能存储系统托管高性能关键数据、同时将较冷的数据重新定位到对象存储系统、包括私有云和公有云选项。

由于元数据是集中的、因此易管理性不会受到影响、您可以照常查看所有数据。ONTAP FabricPool 会自动管理底层数据块的位置。

从何处查找其他信息

如需详细了解本文档所述的信息，请参见以下文档和/或网站：

- TR-4598 FabricPool 最佳实践
<http://www.netapp.com/us/media/tr-4598.pdf>
- TR-3633: 基于 ONTAP 的 Oracle
<http://www.netapp.com/us/media/tr-3633.pdf>
- TR-4591: 数据库数据保护
<http://www.netapp.com/us/media/tr-4591.pdf>
- TR-4592: 基于 MetroCluster 的 Oracle
<http://www.netapp.com/us/media/tr-4592.pdf>
- TR-4534 : 将 Oracle 数据库迁移到 NetApp 存储系统
<http://www.netapp.com/us/media/tr-4534.pdf>

版本历史

版本	日期	文档版本历史
1.0 版	2018 年 4 月	初始版本
1.2 版	2021年4月	支持 ONTAP 9.8。

要验证您的特定环境是否支持本文档所述的确切产品和功能版本，请参见 NetApp 支持站点上的[互操作性表工具 \(IMT\)](#)。NetApp IMT 中定义的产品组件和版本可用于构建 NetApp 所支持的配置。具体的配置结果取决于每个客户如何依照所发布规格进行安装。

版权信息

版权所有 © 2021 NetApp, Inc. 保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

本文档中所含数据与商用项目（按照 FAR 2.101 中的定义）相关，属于 NetApp, Inc. 的专有信息。美国政府对这些数据的使用权具有非排他性、不可转让权、无转授权、全球性、受限不可撤销的许可，但仅限于在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b) 条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。

TR-469-0421-CN