Oracle数据保护

Enterprise applications

NetApp
February 11, 2026

目录

Oracle数据保护

利用ONTAP实现数据保护

NetApp知道、数据库中的任务关键型数据最多。

企业无法在不访问其数据的情况下运营、有时数据决定了业务。这些数据必须受到保护；但是、数据保护不仅仅是确保备份可用、它还需要快速可靠地执行备份、同时还要安全地存储这些备份。

数据保护的另一面是数据恢复。如果无法访问数据、则企业会受到影响、并且可能无法运行、直到数据还原为止。此过程必须快速可靠。最后、必须保护大多数数据库免受灾难的影响、这意味着需要维护数据库的副本。副本必须足够最新。使副本成为一个完全正常运行的数据库还必须快速而简单。

本文档可替代先前发布的技术报告_TR-4591：《Oracle数据保护：备份、恢复和复制》

规划

正确的企业级数据保护架构取决于数据保留、可恢复性以及在各种事件期间对中断的承受能力方面的业务要求。

例如、考虑范围内的应用程序、数据库和重要数据集的数量。为单个数据集构建备份策略以确保符合典型SLA要求相当简单、因为无需管理太多对象。随着数据集数量的增加、监控变得更加复杂、管理员可能不得不花费越来越多的时间来解决备份故障。随着环境达到云和服务提供商规模、需要采用完全不同的方法。

数据集大小也会影响策略。例如、由于数据集非常小、因此对于使用100 GB数据库进行备份和恢复、有许多选项可供选择。只需使用传统工具从备份介质中复制数据、通常就能提供足够的恢复回路(Recovery)。100 TB数据库通常需要完全不同的策略、除非RTO允许发生多天中断、在这种情况下、可以使用基于副本的传统备份和恢复操作步骤。

最后、备份和恢复过程本身之外还有其他因素。例如、是否存在支持关键生产活动的数据库、从而使恢复成为仅由熟练的数据库管理人员执行的罕见事件？或者、数据库是否属于一个大型开发环境、在该环境中、恢复频繁发生、并由一个通才型IT团队进行管理?

RTO、RPO和SLA规划

借助ONTAP、您可以根据业务需求轻松定制Oracle数据库数据保护策略。

这些要求包括恢复速度、允许的最大数据丢失量以及备份保留需求等因素。数据保护计划还必须考虑数据保留和还原方面的各种法规要求。最后、必须考虑不同的数据恢复场景、从因用户或应用程序错误而导致的典型和可预见的恢复到包括站点完全丢失在内的灾难恢复场景。

对数据保护和恢复策略进行微小更改可能会对存储、备份和恢复的整体架构产生显著影响。在开始设计工作之前、必须定义并记录标准、以避免使数据保护架构复杂化。不必要的功能或保护级别会导致不必要的成本和管理开销、而最初被忽视的要求可能会导致项目方向错误或需要在最后一刻更改设计。

恢复时间目标

恢复时间目标(Recovery Time目标、Recovery Time目标、Recovery Time目标、Recovery Time目标、Recovery Time目标、Recovery Time目标)定义了恢复服务所允许的最长时间。例如、人力资源数据库的RTO可能为24小时、因为虽然在工作日无法访问此数据会非常不便、但业务仍可继续运营。相比之下、支持

银行总分类账的数据库将以分钟甚至几秒钟计量的最短时间。RTO不可能为零、因为必须有方法区分实际服务中断和例行事件(例如网络数据包丢失)。但是、RTO接近零是一项典型要求。

恢复点目标

恢复点目标(RPO)定义了可容忍的最大数据丢失。在许多情况下、RPO完全取决于快照或SnapMirror更新的频率。

在某些情况下、可以通过更频繁地有选择地保护某些数据来提高RPO的主动性。在数据库环境中、RPO通常是指在特定情况下可能丢失多少日志数据的问题。在典型的恢复情形中、如果数据库因产品错误或用户错误而损坏、则RPO应为零、这意味着不会丢失任何数据。恢复操作步骤包括还原数据库文件的早期副本、然后重影日志文件、以使数据库状态达到所需的时间点。此操作所需的日志文件应已位于原始位置。

在异常情况下、日志数据可能会丢失。例如、意外事件或恶意事件 `rm -rf *` 数据库文件的数量可能会导致所有数据被删除。唯一的选择是从备份(包括日志文件)进行还原、而某些数据将不可避免地丢失。在传统备份环境中、要提高RPO、唯一的选择是对日志数据执行重复备份。但是、由于数据会不断移动、而且很难将备份系统作为一项持续运行的服务来维护、因此这一点存在一些限制。高级存储系统的优势之一是能够保护数据免受文件意外或恶意损坏、从而提供更好的RPO、而无需移动数据。

灾难恢复

灾难恢复包括在发生物理灾难时恢复服务所需的IT架构、策略和过程。这可能包括洪水、火灾或有恶意或疏忽意图的人员。

灾难恢复不仅仅是一组恢复过程。它是一个完整的过程、可以识别各种风险、定义数据恢复和服务连续性要求、并提供具有相关过程的正确架构。

在确定数据保护要求时、必须区分典型的RPO和RTO要求以及灾难恢复所需的RPO和RTO要求。某些应用程序环境要求RPO为零、RTO接近零、以应对从相对正常的用户错误到破坏数据中心的火灾等数据丢失情形。然而，这种高水平的保护会产生成本和行政后果。

通常、非灾难数据恢复要求应严格、原因有两个。首先、破坏数据的应用程序错误和用户错误是可以预见的、几乎是不可避免的。其次、只要存储系统不被销毁、设计一个能够实现零RPO和低RTO的备份策略不难。没有理由不解决容易补救的重大风险、这就是为什么本地恢复的RPO和RTO目标应该积极主动的原因。

根据发生灾难的可能性以及相关数据丢失或业务中断的后果、灾难恢复RTO和RPO要求的差别更大。RPO和RTO要求应基于实际业务需求、而不是一般原则。它们必须考虑多种逻辑和物理灾难情形。

逻辑灾难

逻辑灾难包括由用户、应用程序或操作系统错误以及软件故障导致的数据损坏。逻辑灾难还可能包括外部人员利用病毒或蠕虫或利用应用程序漏洞进行的恶意攻击。在这些情况下、物理基础架构未损坏、但底层数据不再有效。

一种日益常见的逻辑灾难类型称为勒索软件、在这种情况下、攻击向量用于对数据进行加密。加密不会损坏数据、但在向第三方付款之前、加密将使数据不可用。越来越多的企业正成为勒索软件黑客攻击的专门目标。针对这种威胁、NetApp提供防篡改快照、在这些快照中、即使存储管理员也无法在配置的到期日期之前更改受保护的数据。

物理灾难

物理灾难包括基础架构的组件发生故障、导致其冗余能力超出范围、从而导致数据丢失或服务长时间丢失。例如、RAID保护可提供磁盘驱动器冗余、而使用HBA可提供FC端口和FC缆线冗余。此类组件的硬件故障是可以

预见的、不会影响可用性。

在企业环境中、通常可以使用冗余组件保护整个站点的基础架构、直到唯一可预见的物理灾难情形是站点完全丢失。灾难恢复规划则取决于站点到站点复制。

同步和异步数据保护

理想情况下、所有数据都会在地理位置分散的站点之间同步复制。由于以下几个原因、此类复制并不总是可行甚至不可能实现：

- 同步复制不可避免地会增加写入延迟、因为必须先将所有更改复制到这两个位置、然后应用程序/数据库才能继续处理。所产生的性能影响有时是不可接受的、从而排除了使用同步镜像的可能性。
- 随着100% SSD存储的采用率不断提高、更有可能注意到额外的写入延迟、因为性能预期包括数十万次IOPS和亚微秒延迟。要充分发挥使用100% SSD的优势、可能需要重新审视灾难恢复策略。
- 数据集的字节数持续增长、在确保足够的带宽来支持同步复制方面面临着挑战。
- 数据集的复杂性也在不断增加、在管理大规模同步复制方面也面临着挑战。
- 基于云的策略通常涉及更长的复制距离和延迟、进一步排除了同步镜像的使用。

NetApp提供的解决方案既包括可满足最严苛数据恢复需求的同步复制、也包括可提高性能和灵活性的异步解决方案。此外、NetApp技术还可以与许多第三方复制解决方案(例如Oracle DataGuard)无缝集成

保留时间

数据保护策略的最后一个方面是数据保留时间、数据保留时间可能差别很大。

- 通常要求在主站点上执行14天的夜间备份、在二级站点上执行90天的备份。
- 许多客户创建独立的季度归档、存储在不同的介质上。
- 不断更新的数据库可能不需要历史数据、备份只需保留几天。
- 根据法规要求、可能需要在365天内恢复到任意事务的时间点。

数据库可用性

ONTAP旨在最大程度地提高Oracle数据库的可用性。本文档不会介绍完整的ONTAP高可用性功能问题描述。但是、与数据保护一样、在设计数据库基础架构时、基本了解此功能非常重要。

HA 对

高可用性的基本单位是HA对。每个对都包含冗余链路、以支持将数据复制到NVRAM。NVRAM不是写入缓存。控制器中的RAM用作写入缓存。NVRAM的用途是临时记录数据、以防止发生意外系统故障。在这方面、它类似于数据库重做日志。

NVRAM和数据库重做日志均用于快速存储数据、从而可以尽快提交对数据的更改。直到稍后在ONTAP和大多数数据库平台上的一个称为检查点的过程中、才会更新驱动器(或数据文件)上的永久性数据。在正常操作期间、不会读取NVRAM数据和数据库重做日志。

如果控制器突然出现故障、NVRAM中可能会存储一些尚未写入驱动器的待处理更改。配对控制器会检测到故

障、控制驱动器并应用NVRAM中存储的所需更改。

接管和交还

接管和交还是指在HA对中的节点之间转移存储资源职责的过程。接管和返回有两个方面：

- 管理允许访问驱动器的网络连接
- 驱动器本身的管理

支持CIFS和NFS流量的网络接口配置了主位置和故障转移位置。接管包括将网络接口移动到与原始位置位于同一子网的物理接口上的临时主端口。交还包括将网络接口移回其原始位置。可以根据需要调整确切的行为。

在接管和回放期间、不会重新定位支持iSCSI和FC等SAN块协议的网络接口。而是应使用包含完整HA对的路径来配置LUN、从而生成主路径和二级路径。

> 此外、还可以配置指向其他控制器的其他路径、以支持在较大集群中的节点之间重新定位数据、但这不是HA过程的一部分。

接管和返回的第二个方面是磁盘所有权的传输。具体过程取决于多个因素、包括接管/还原的原因以及发出的命令行选项。目标是尽可能高效地执行操作。虽然整个过程看起来可能需要几分钟时间、但驱动器所有权从一个节点转换到另一个节点的实际时刻通常可以以秒为单位进行衡量。

接管时间

在接管和备份操作期间、主机I/O会短暂暂停、但在配置正确的环境中、不应发生应用程序中断。I/O延迟的实际过渡过程通常以秒为单位、但主机可能需要更多时间来识别数据路径中的更改并重新提交I/O操作。

中断的性质取决于协议：

- 在过渡到新物理位置后、支持NFS和CIFS流量的网络接口会向网络发出地址解析协议(Address Resolution Protocol、ARP)请求。这会导致网络交换机更新其介质访问控制(MAC)地址表并恢复处理I/O在计划内接管和移交的情况下、中断通常以秒为单位进行衡量、在许多情况下、无法检测到。某些网络可能较慢、无法完全识别网络路径的变化、而某些操作系统可能会在很短的时间内排队等待大量I/O、必须重试。这会延长恢复I/O所需的时间
- 支持SAN协议的网络接口不会过渡到新位置。主机操作系统必须更改正在使用的一个或多个路径。主机观察到的I/O暂停取决于多个因素。从存储系统角度来看、无法提供I/O的时间段仅为几秒。但是、不同的主机操作系统可能需要额外的时间才能使I/O在重试之前超时。较新的操作系统能够更快地识别路径更改、但较旧的操作系统通常需要长达30秒才能识别更改。

下表显示了存储系统无法为应用程序环境提供数据的预期接管时间。在任何应用程序环境中都不应出现任何错误、接管应显示为IO处理中的短暂暂停。

	NFS	AFF	ASA
计划内接管	15秒	第个问题	2-3秒
计划外接管	30秒	第个问题	2-3秒

校验和与数据完整性

ONTAP及其支持的协议包括多项功能、可保护Oracle数据库完整性、包括空闲数据和通过网络传输的数据。

ONTAP中的逻辑数据保护包括三个关键要求：

- 必须防止数据损坏。
- 必须保护数据免受驱动器故障的影响。
- 必须防止对数据所做的更改丢失。

以下各节将讨论这三种需求。

网络损坏：校验和

最基本的数据保护级别是校验和、校验和是随数据一起存储的一种特殊错误检测代码。使用校验和(在某些情况下、使用多个校验和)检测网络传输期间的数据损坏。

例如、FC帧包含一种称为循环冗余校验(CRC)的校验和形式、用于确保有效负载在传输过程中不会损坏。发射器会同时发送数据和数据的CRC。FC帧的接收器重新计算已接收数据的CRC、以确保其与已传输的CRC匹配。如果新计算的CRC与附加到帧的CRC不匹配、则数据将损坏、FC帧将被丢弃或拒绝。iSCSI I/O操作包括TCP/IP和以太网层的校验和、并且为了提供额外保护、还可以在SCSI层提供可选的CRC保护。TCP层或IP层会检测到线路上的任何位损坏、从而导致数据包重新传输。与FC一样、SCSI CRC中的错误会导致丢弃或拒绝操作。

驱动器损坏：校验和

校验和还用于验证存储在驱动器上的数据的完整性。写入驱动器的数据块使用校验和功能进行存储、该功能会产生与原始数据相关的不可预测的数字。从驱动器中读取数据时、将重新计算校验和并将其与存储的校验和进行比较。如果不匹配、则数据已损坏、必须由RAID层进行恢复。

数据损坏：写入丢失

最难检测的损坏类型之一是写入丢失或放错位置。确认写入后、必须将其写入到正确位置的介质中。通过使用随数据存储的简单校验和、可以相对容易地检测原位数据损坏。但是、如果只是写入丢失、则先前版本的数据可能仍存在、校验和将是正确的。如果将写入放置在错误的物理位置、则关联的校验和将再次对存储的数据有效、即使写入操作已销毁其他数据。

应对此挑战的解决方案如下：

- 写入操作必须包含元数据、用于指示预期写入位置。
- 写入操作必须包含某种版本标识符。

当ONTAP写入块时、它会包含有关块所属位置的数据。如果后续读取发现某个块、但在位置456发现元数据时、元数据指示该块属于位置123、则表示该写入已放错位置。

检测完全丢失的写入操作会更加困难。解释非常复杂、但从本质上说、ONTAP存储元数据的方式是、写入操作会导致更新到驱动器上的两个不同位置。如果写入丢失、则后续读取的数据和关联元数据将显示两个不同的版本标识。这表示驱动器未完成写入。

丢失和放错位置的写入损坏极为少见、但随着驱动器不断增长、数据集逐渐扩展到EB级、风险也会增加。支持数据库工作负载的任何存储系统都应包括失写检测。

驱动器故障：RAID、RAID DP和RAID-TEC

如果发现驱动器上的数据块已损坏、或者整个驱动器发生故障且完全不可用、则必须重新生成数据。这在ONTAP中通过使用奇偶校验驱动器来实现。数据在多个数据驱动器之间进行条带化、然后生成奇偶校验数据。该数据与原始数据分开存储。

ONTAP最初使用的是RAID 4、该RAID 4会为每组数据驱动器使用一个奇偶校验驱动器。这样、组中的任何一个驱动器都可能发生故障、而不会导致数据丢失。如果奇偶校验驱动器发生故障、则不会损坏任何数据、可以构建新的奇偶校验驱动器。如果一个数据驱动器发生故障、则其余驱动器可与奇偶校验驱动器结合使用来重新生成缺失的数据。

如果驱动器较小、则两个驱动器同时发生故障的统计几率可以忽略不计。随着驱动器容量的增长、在驱动器发生故障后重建数据所需的时间也会相应增加。这增加了第二个驱动器故障导致数据丢失的时间范围。此外、重建过程会在无故障驱动器上创建大量额外的I/O。随着驱动器老化、导致第二个驱动器故障的额外负载风险也会增加。最后、即使持续使用RAID 4不会增加数据丢失的风险、数据丢失的后果也会更加严重。RAID组发生故障时丢失的数据越多、恢复数据所需的时间就越长、从而延长业务中断时间。

这些问题促使NetApp开发了NetApp RAID DP技术、这是RAID 6的变体。此解决方案包含两个奇偶校验驱动器、这意味着RAID组中的任何两个驱动器都可能发生故障、而不会造成数据丢失。驱动器的大小持续增长、这最终导致NetApp开发了NetApp RAID-TEC技术、该技术引入了第三个奇偶校验驱动器。

一些历史数据库最佳实践建议使用RAID-10、也称为条带化镜像。这提供的数据保护比RAID DP更少、因为存在多种双磁盘故障情形、而在RAID DP中则没有。

还有一些历史数据库最佳实践表明、出于性能考虑、RAID-10优于RAID-4/5/6选项。这些建议有时会提及RAID惩罚。虽然这些建议通常是正确的、但不适用于在ONTAP中实施RAID。性能问题与奇偶校验重新生成有关。在传统RAID实施中、处理数据库执行的例行随机写入需要多次磁盘读取才能重新生成奇偶校验数据并完成写入。惩罚定义为执行写入操作所需的额外读取IOPS。

ONTAP不会产生RAID惩罚、因为写入会暂存到内存中、在该内存中会生成奇偶校验、然后作为单个RAID条带写入磁盘。完成写入操作不需要执行任何读取操作。

总之、与RAID 10相比、RAID DP和RAID-TEC可提供更多的可用容量、更好地防止驱动器故障、并且不会影响性能。

硬件故障保护：NVRAM

任何为数据库工作负载提供服务的存储阵列都必须尽快为写入操作提供服务。此外、必须保护写入操作、使其不会因意外事件(例如断电)而丢失。这意味着任何写入操作都必须安全地存储在至少两个位置。

AFF和FAS系统依靠NVRAM来满足这些要求。写入过程的工作原理如下：

1. 入站写入数据存储在RAM中。
2. 必须对磁盘上的数据所做的更改会记录到本地节点和配对节点上的NVRAM中。NVRAM不是写入缓存、而是类似于数据库重做日志的日志。在正常情况下、不会读取它。它仅用于恢复、例如在I/O处理期间发生电源故障后。
3. 然后、写入操作会向主机确认。

从应用程序角度来看、此阶段的写入过程已完成、数据会受到保护、不会丢失、因为数据会存储在两个不同的位

置。更改最终会写入磁盘、但从应用程序角度来看、此过程是带外过程、因为它发生在确认写入之后、因此不会影响延迟。此过程再次类似于数据库日志记录。对数据库所做的更改会尽快记录在重做日志中、然后确认已提交更改。数据文件的更新发生得更晚、不会直接影响处理速度。

如果某个控制器发生故障、配对控制器将接管所需磁盘的所有权、并在NVRAM中回显已记录的数据、以恢复发生故障时正在进行的任何I/O操作。

硬件故障保护：**NVFAIL**

如前文所述、写入操作只有在至少另一个控制器上记录到本地NVRAM和NVRAM后才会得到确认。此方法可确保硬件故障或断电不会导致传输中I/O丢失如果本地NVRAM发生故障或与HA配对节点的连接发生故障、则不会再镜像此传输中的数据。

如果本地NVRAM报告错误、则此节点将关闭。此关闭会导致故障转移到HA配对控制器。由于发生故障的控制器尚未确认写入操作、因此不会丢失任何数据。

除非强制执行故障转移、否则ONTAP不允许在数据不同步时进行故障转移。以这种方式强制更改条件即表示数据可能会留在原始控制器中、并且数据丢失是可以接受的。

如果强制执行故障转移、数据库尤其容易受到损坏的影响、因为数据库会在磁盘上保留大量内部数据缓存。如果发生强制故障转移、先前确认的更改将被有效丢弃。存储阵列的内容会及时有效地向后跳转、数据库缓存的状态不再反映磁盘上数据的状态。

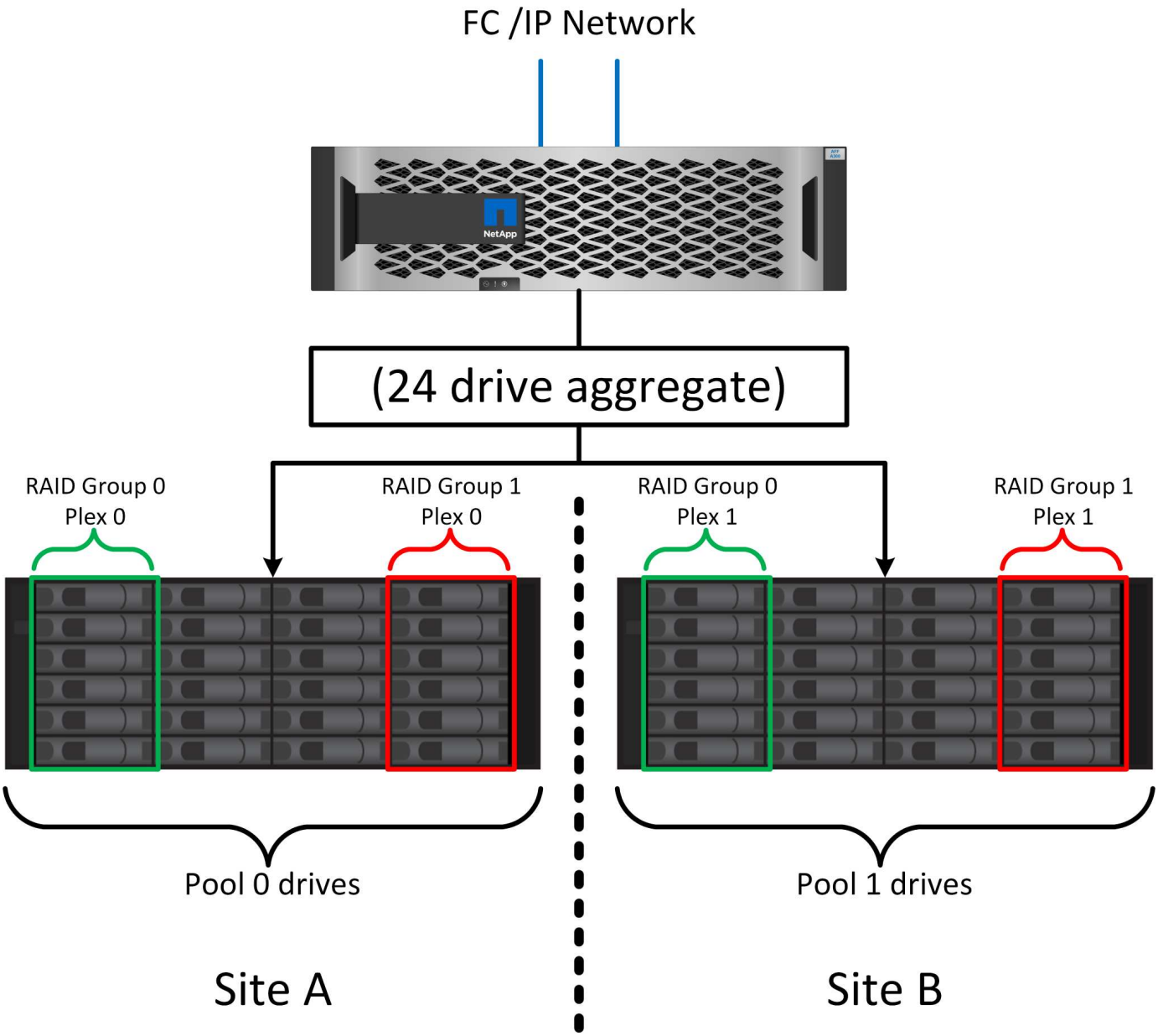
为了保护数据免受这种情况的影响、ONTAP允许对卷进行配置、以便针对NVRAM故障提供特殊保护。触发此保护机制后、卷将进入名为NVFAIL的状态。此状态会导致I/O错误、即发生原因A应用程序会关闭、以使其不使用陈旧数据。数据不应丢失、因为存储阵列上应存在任何已确认的写入。

通常的后续步骤是、管理员先完全关闭主机、然后再手动将LUN和卷重新联机。虽然这些步骤可能涉及一些工作、但这种方法是确保数据完整性的最安全方法。并非所有数据都需要这种保护、这就是可以逐个卷配置NVFAIL行为的原因。

站点和磁盘架故障保护：**SyncMirror**和**plexes**

SyncMirror是一种镜像技术、可增强但不会取代RAID DP或RAID-TEC。它会镜像两个独立RAID组的内容。逻辑配置如下：

- 驱动器会根据位置配置到两个池中。一个池由站点A上的所有驱动器组成、另一个池由站点B上的所有驱动器组成
- 然后、基于RAID组的镜像集创建一个通用存储池(称为聚合)。从每个站点提取的驱动器数量相等。例如、一个包含20个驱动器的SyncMirror聚合将由站点A的10个驱动器和站点B的10个驱动器组成
- 给定站点上的每组驱动器都会自动配置为一个或多个完全冗余的RAID-DP或RAID-TEC组、而与镜像的使用无关。这样可以提供持续的数据保护、即使在站点丢失后也是如此。



上图显示了一个示例SyncMirror配置。在控制器上创建了一个包含24个驱动器的聚合、其中12个驱动器来自站点A上分配的磁盘架、12个驱动器来自站点B上分配的磁盘架这些驱动器被分组为两个镜像RAID组。RAID组0在站点A上包含一个6驱动器丛、该丛镜像到站点B上的6驱动器丛同样、RAID组1在站点A上包含一个6驱动器丛、该丛镜像到站点B上的6驱动器丛

SyncMirror通常用于为MetroCluster系统提供远程镜像、每个站点有一个数据副本。有时、它会用于在单个系统中提供额外的冗余级别。尤其是、它可以提供磁盘架级冗余。驱动器架已包含双电源和控制器、总体比金属板稍多、但在某些情况下、可能需要额外保护。例如、一家NetApp客户为汽车测试期间使用的移动实时分析平台部署了SyncMirror。该系统分为两个物理机架、由独立UPS系统的独立电源供电。

校验和

对于习惯于使用Oracle RMAN流式备份并迁移到基于快照的备份的数据库用户来说、校验和主题特别重要。RMAN的一项功能是、它会在备份操作期间执行完整性检查。尽管此功能具有一定的价值、但其主要优势是用于未在现代存储阵列上使用的数据库。将物理驱动器用于Oracle数据库时、几乎可以肯定、随着驱动器老化、最终会发生损坏、而在真正的存储阵列中、基于阵列的校验和可以解决这一问题。

对于真正的存储阵列、数据完整性可通过在多个级别使用校验和来保护。如果基于IP的网络中的数据损坏、则传输控制协议(TCP)层会拒绝数据包数据并请求重新传输。FC协议包括校验和、封装的SCSI数据也是如此。将ONTAP置于阵列上后、它将具有RAID和校验和保护功能。可能会发生损坏、但与大多数企业阵列一样、系统会检测到并更正此问题。通常、整个驱动器发生故障、提示重建RAID、数据库完整性不受影响。驱动器上的单个字节仍然可能被宇宙辐射或闪存单元故障损坏。如果发生这种情况、奇偶校验检查将失败、驱动器将出现故障、并开始RAID重建。同样、数据完整性也不受影响。最后一道防线是使用校验和。例如、如果驱动器损坏的数据出现灾难性固件错误、而RAID奇偶校验检查无法检测到该数据、则校验和将不匹配、ONTAP将阻止在Oracle数据库收到损坏的块之前传输该块。

Oracle数据文件和重做日志架构还旨在提供尽可能高级别的数据完整性、即使在极端情况下也是如此。在最基本的层面上、Oracle块包括对几乎每个I/O进行校验和和和基本逻辑检查如果Oracle未崩溃或使表空间脱机、则数据完好无损。数据完整性检查的程度可以调整、Oracle也可以配置为确认写入。因此、几乎所有崩溃和故障情形都可以恢复、在极少数情况下发生不可恢复的情况时、系统会立即检测到损坏。

大多数使用Oracle数据库的NetApp客户在迁移到基于快照的备份之后不再使用RMAN和其他备份产品。在使用SnapCenter执行块级恢复时、仍然可以使用RMAN。但是、在日常工作中、RMAN、NetBackup和其他产品仅偶尔用于创建每月或每季度归档副本。

有些客户选择运行 `dbv` 定期对其现有数据库执行完整性检查。NetApp不建议采用这种做法、因为它会产生不必要的I/O负载。如上所述、如果数据库之前未遇到问题、则可能会出现 `dbv` 检测问题几乎为零、此实用程序会在网络和存储系统上创建非常高的顺序I/O负载。除非有理由认为存在损坏、例如暴露于已知的Oracle错误、否则没有理由运行 `dbv`。

备份和恢复基础知识

基于**Snapshot**的备份

基于ONTAP的Oracle数据库数据保护的基础是NetApp Snapshot技术。

关键值如下：

- *精简性。*快照是指特定时间点数据容器内容的只读副本。
- *效率。*创建快照时不需要任何空间。只有在数据发生更改时才会占用空间。
- *易管理性。*基于快照的备份策略易于配置和管理、因为快照是存储操作系统的本机部分。如果存储系统已启动、则它可以随时创建备份。
- *可扩展性。*一个文件和LUN容器最多可保留1024个备份。对于复杂的数据集、可以通过一组一致的快照来保护多个数据容器。
- 无论卷包含1024个快照还是无快照、性能都不受影响。

虽然许多存储供应商都提供快照技术、但ONTAP中的快照技术是独一无二的、可为企业级应用程序和数据库环境带来显著优势：

- Snapshot副本是底层任意位置写入文件布局(Write-Anywhere File Layout、WAFL)的一部分。它们不是附加技术或外部技术。由于存储系统是备份系统、因此可简化管理。
- Snapshot副本不会影响性能、但某些边缘情形除外、例如、快照中存储的数据量如此之多、以致于底层存储系统会填满。
- 术语"一致性组"通常用于指作为一致的数据集合进行管理的一组存储对象。特定ONTAP卷的快照构成一致性组备份。

ONTAP快照的扩展能力也优于竞争技术。客户可以存储5个、50个或500个快照、而不会影响性能。卷中当前允许的最大快照数为1024。如果需要额外保留快照、可以选择将快照级联到其他卷。

因此、保护ONTAP上托管的数据集非常简单、并且具有高度可扩展性。备份不需要移动数据、因此可以根据业务需求定制备份策略、而不是网络传输速率、大量磁带驱动器或磁盘暂存区的限制。

快照是否为备份?

有关将快照用作数据保护策略的一个常见问题是、"实际"数据和快照数据位于同一个驱动器上。丢失这些驱动器将导致主数据和备份均丢失。

这是一个合理的问题。本地快照用于满足日常备份和恢复需求、在这方面、快照是备份。在NetApp环境中、几乎99%的恢复方案都依靠快照来满足最苛刻的恢复时间要求。

但是、本地快照绝不是唯一的备份策略、这就是NetApp提供SnapMirror和SnapVault复制等技术来快速高效地将快照复制到一组独立驱动器的原因。在采用快照和快照复制功能且架构合理的解决方案中、磁带的使用量可以降至最低、甚至可以每季度归档一次、也可以完全避免。

基于**Snapshot**的备份

使用ONTAP Snapshot副本保护数据有多种选择、快照是复制、灾难恢复和克隆等许多其他ONTAP功能的基础。本文档不会介绍有关Snapshot技术的完整问题描述、但以下各节将提供一般概述。

创建数据集快照的主要方法有两种：

- 崩溃状态一致的备份
- 应用程序一致的备份

崩溃状态一致的数据集备份是指在一个时间点捕获整个数据集结构。如果数据集存储在单个卷中、则此过程非常简单；可以随时创建Snapshot。如果数据集跨越多个卷、则必须创建一致性组(CG)快照。创建CG快照的选项有多种、包括NetApp SnapCenter软件、本机ONTAP一致性组功能以及用户维护的脚本。

当备份点恢复足以满足要求时、主要使用崩溃状态一致的备份。当需要更精细的恢复时、通常需要应用程序一致的备份。

"应用程序一致"中的"一致"一词通常用词不当。例如、将Oracle数据库置于备份模式称为应用程序一致的备份、但数据不会以任何方式保持一致或处于静态。数据在整个备份过程中持续更改。相比之下、大多数MySQL和Microsoft SQL Server备份确实会在执行备份之前将数据置于静噪状态。VMware可能会使某些文件保持一致、也可能不会使其保持一致。

一致性组

术语"一致性组"是指存储阵列能够将多个存储资源作为一个映像进行管理。例如、一个数据库可能包含10个LUN。阵列必须能够以一致的方式备份、还原和复制这10个LUN。如果LUN的映像在备份时不一致、则无法还原。复制这10个LUN要求所有副本之间完全同步。

在讨论ONTAP时、不经常使用术语"一致性组"、因为一致性一直是ONTAP中卷和聚合架构的基本功能。许多其他存储阵列将LUN或文件系统作为单独的单元进行管理。然后、可以选择将其配置为"一致性组"以实现数据保护、但这是配置中的一个额外步骤。

ONTAP始终能够捕获一致的本地和复制数据映像。虽然ONTAP系统上的各种卷通常不会正式描述为一致性组、但它们就是一致性组。该卷的快照是一致性组映像、该快照的还原是一致性组还原、SnapMirror和SnapVault均

提供一致性组复制。

一致性组快照

一致性组快照(CG快照)是基本ONTAP快照技术的扩展。标准快照操作会为单个卷中的所有数据创建一致的映像、但有时需要在多个卷甚至多个存储系统之间创建一组一致的快照。这样就会生成一组快照、这些快照的使用方式与仅包含一个卷的快照相同。它们可用于本地数据恢复、为灾难恢复目的进行复制或作为一个一致的单元进行克隆。

已知的最大CG-Snapsh图 用途是用于大小约为1 PB且跨越12个控制器的数据库环境。在此系统上创建的CG快照已用于备份、恢复和克隆。

大多数情况下、如果数据集跨越多个卷且必须保留写入顺序、则选定管理软件会自动使用CG快照。在这种情况下、无需了解CG快照的技术详细信息。但是、在某些情况下、复杂的数据保护要求需要对数据保护和复制过程进行详细控制。可以选择自动化工作流或使用自定义脚本来调用CG-Snapshot API。要了解cG-Snapshot的最佳选项和角色、需要对该技术进行更详细的说明。

创建一组CG快照的过程分为两步：

1. 在所有目标卷上建立写入隔离。
2. 在隔离状态下创建这些卷的快照。

写入隔离是按序列建立的。这意味着、在多个卷之间设置隔离过程时、写入I/O会冻结在序列中的第一个卷上、因为它会继续提交到稍后显示的卷。最初、这可能看起来违反了保留写入顺序的要求、但只有在主机上异步发出的适用场景I/O、而不依赖于任何其他写入。

例如、数据库可能会对大量异步数据文件更新进行问题描述、并允许操作系统重新排列I/O、然后根据自己的计划程序配置完成这些更新。无法保证此类I/O的顺序、因为应用程序和操作系统已释放保留写入顺序的要求。

作为一个计数器示例、大多数数据库日志记录活动都是同步的。在确认I/O并保留这些写入顺序之前、数据库不会继续进行日志写入。如果日志I/O到达隔离的卷、则不会进行确认、应用程序会阻止进一步写入。同样、文件系统元数据I/O通常是同步的。例如、文件删除操作不能丢失。如果带有xfs文件系统的操作系统删除了某个文件、而更新了xfs文件系统元数据以删除对该文件的引用的I/O则会登录到隔离的卷上、则文件系统活动将暂停。这样可以保证CG快照操作期间文件系统的完整性。

在目标卷之间设置写入隔离后、这些卷便可创建快照了。无需同时创建快照、因为从依赖写入的角度来看、卷的状态是冻结的。为了防止创建CG快照的应用程序出现缺陷、初始写入隔离包括一个可配置的超时时间、在此超时时间内、ONTAP会自动释放隔离并在定义的秒数后恢复写入处理。如果所有快照都是在超时期限到期之前创建的、则生成的一组快照是有效的一致性组。

从属写入顺序

从技术角度来看、一致性组的关键在于保留写入顺序、尤其是依赖写入顺序。例如、向10个LUN写入数据的数据库会同时向所有LUN写入数据。许多写入操作是异步发出的、这意味着它们的完成顺序并不重要、实际完成顺序会因操作系统和网络行为而异。

在数据库继续执行其他写入操作之前、磁盘上必须存在某些写入操作。这些关键写入操作称为依赖写入。后续写入I/O取决于磁盘上是否存在这些写入。对这10个LUN执行任何快照、恢复或复制操作都必须确保依赖写入顺序得到保证。文件系统更新是依赖写入顺序写入的另一个示例。必须保留文件系统更改的顺序、否则整个文件系统可能会损坏。

战略

基于快照的备份有两种主要方法：

- 崩溃状态一致的备份
- 受Snapshot保护的热备份

崩溃状态一致的数据库备份是指在一个时间点捕获整个数据库结构、包括数据文件、重做日志和控制文件。如果数据库存储在单个卷中、则此过程非常简单；可以随时创建Snapshot。如果数据库跨越多个卷、则必须创建一致性组(CG)快照。创建CG快照的选项有多种、包括NetApp SnapCenter软件、本机ONTAP一致性组功能以及用户维护的脚本。

崩溃状态一致的Snapshot备份主要在备份点恢复已足够时使用。在某些情况下、可以应用归档日志、但在需要更精细的时间点恢复时、最好使用联机备份。

基于快照的联机备份的基本操作步骤如下所示：

1. 将数据库放置在中 `backup` 模式。
2. 为托管数据文件的所有卷创建快照。
3. 退出 `backup` 模式。
4. 运行命令 `alter system archive log current` 强制日志归档。
5. 为托管归档日志的所有卷创建快照。

此操作步骤将生成一组快照、其中包含处于备份模式的数据文件以及处于备份模式时生成的关键归档日志。这是恢复数据库的两项要求。为方便起见、还应保护控制文件等文件、但唯一的绝对要求是保护数据文件和归档日志。

虽然不同的客户可能有非常不同的策略、但几乎所有这些策略最终都基于下面所述的相同原则。

基于**Snapshot**的恢复

在为Oracle数据库设计卷布局时、首先要决定是否使用基于卷的NetApp SnapRestore (VBSR)技术。

基于卷的SnapRestore可以将卷几乎即时还原到较早的时间点。由于卷上的所有数据均已还原、因此VBSR可能并不适用于所有使用情形。例如、如果整个数据库(包括数据文件、重做日志和归档日志)存储在单个卷上、而此卷通过VBSR还原、则数据会丢失、因为较新的归档日志和重做数据会被丢弃。

还原不需要VBSR。许多数据库都可以通过使用基于文件的单文件文件系统(Single File SnapRestore、SFSR)进行还原、或者只需将文件从快照复制回活动文件系统即可。

当数据库非常大或必须尽快恢复时、最好使用VBSR、而使用VBSR需要隔离数据文件。在NFS环境中、给定数据库的数据文件必须存储在未受任何其他类型文件污染的专用卷中。在SAN环境中、数据文件必须存储在专用卷上的专用LUN中。如果使用卷管理器(包括Oracle自动存储管理[ASM])、则磁盘组还必须专用于数据文件。

通过以这种方式隔离数据文件、可以将其还原到早期状态、而不会损坏其他文件系统。

Snapshot 预留

对于SAN环境中包含Oracle数据的每个卷、 `percent-snapshot-space` 应设置为零、因为在LUN环境中为快照预留空间没有用处。如果预留百分比设置为100、则包含LUN的卷的快照需要该卷中具有足够的可用空间(不包

括快照预留)来吸收所有数据的100%周转率。如果预留百分比设置为较低的值、则所需的可用空间量相应较少、但始终不包括Snapshot预留。这意味着会浪费LUN环境中的快照预留空间。

在NFS环境中、有两种选择：

- 设置 `percent-snapshot-space` 基于预期的Snapshot空间消耗。
- 设置 `percent-snapshot-space` 将活动空间和快照空间占用情况统一置零并进行管理。

使用第一个选项时、 `percent-snapshot-space` 设置为非零值、通常约为20%。然后、此空间将对用户隐藏。但是、此值不会对利用率造成限制。如果预留百分比为20%的数据库的周转率为30%、则快照空间可能会超出预留百分比的界限并占用未预留空间。

将预留设置为20%这样的值的主要优势是、验证某些空间始终可用于快照。例如、预留为20%的1 TB卷仅允许数据库管理员(Database Administrator、DBA)存储800 GB数据。此配置可确保至少为快照占用200 GB的空间。

时间 `percent-snapshot-space` 设置为零时、卷中的所有空间均可供最终用户使用、从而提高可见性。数据库管理员必须了解、如果发现1 TB卷利用快照、则这1 TB空间将在活动数据和Snapshot周转率之间共享。

最终用户之间没有明确的首选方案一和备选方案二。

ONTAP和第三方快照

Oracle文档ID 604683.1介绍了第三方快照支持的要求以及可用于备份和还原操作的多个选项。

第三方供应商必须保证公司的快照符合以下要求：

- 快照必须与Oracle建议的还原和恢复操作集成。
- 快照必须在快照点保持数据库崩溃状态一致。
- 系统会为快照中的每个文件保留写入顺序。

ONTAP和NetApp Oracle管理产品符合这些要求。

SnapRestore

NetApp SnapRestore技术可在ONTAP中从快照快速恢复数据。

当关键数据集不可用时、关键业务运营将中断。磁带可能会中断、甚至从基于磁盘的备份中恢复的速度也可能很慢、无法通过网络传输。SnapRestore通过近乎即时地还原数据集来避免这些问题。即使是PB级数据库、也只需几分钟的时间即可完全还原。

SnapRestore有两种形式：基于文件/LUN和基于卷。

- 单个文件或LUN可以在几秒钟内还原、无论它是2 TB LUN还是4 KB文件。
- 文件或LUN容器可以在几秒钟内还原、无论数据大小是10 GB还是100 TB。

"文件或LUN容器"通常指FlexVol卷。例如、一个卷中可能有10个LUN构成一个LVM磁盘组、或者一个卷可能会存储包含1000个用户的NFS主目录。您可以将整个卷作为一个操作来还原、而不是对每个文件或LUN执行还原操作。此过程还适用于包含多个卷的横向扩展容器、例如FlexGroup或ONTAP一致性组。

SnapRestore之所以能够如此快速高效地工作、是因为快照的性质、从本质上说、快照是一个在特定时间点卷内

容的并行只读视图。活动块是可以更改的实际块、而快照是创建快照时构成文件和LUN的块的状态的只读视图。

ONTAP仅允许对快照数据进行只读访问、但可以使用SnapRestore重新激活这些数据。快照将重新启用为数据的读写视图、从而将数据恢复到先前的状态。SnapRestore可以在卷或文件级别运行。该技术本质上是相同的、但行为略有不同。

Volume SnapRestore

基于卷的SnapRestore会将整个数据卷返回到先前的状态。此操作不需要移动数据、这意味着还原过程基本上是瞬时的、尽管处理API或CLI操作可能需要几秒钟时间。还原1 GB的数据并不比还原1 PB的数据更复杂、也不会更耗时。这一功能是许多企业客户迁移到ONTAP存储系统的主要原因。即使是最大的数据集、它也能以秒为单位提供一个RTO。

基于卷的SnapRestore的一个缺点是、卷内的更改会随着时间的推移而累积。因此、每个快照和活动文件数据都取决于到那时为止所做的更改。将卷还原到早期状态意味着、系统将会先对数据进行所有后续更改、然后再进行相应的更改。但是、不太明显的是、这包括随后创建的快照。这并不总是可取的。

例如、数据保留SLA可能指定30天的夜间备份。如果将数据集还原到五天前使用卷SnapRestore创建的快照、则会丢弃前五天创建的所有快照、从而违反SLA。

有许多选项可用于解决此限制：

1. 可以从先前的快照复制数据、而不是对整个卷执行SnapRestore。此方法最适合较小的数据集。
2. 快照可以克隆、而不是还原。此方法的限制是、源快照是克隆的依赖项。因此、除非同时删除克隆或将其拆分成独立的卷、否则无法将其删除。
3. 使用基于文件的SnapRestore。

File SnapRestore

基于文件的SnapRestore是一种基于快照的更精细还原过程。系统会还原单个文件或LUN的状态、而不是还原整个卷的状态。无需删除任何快照、此操作也不会对先前的快照创建任何依赖关系。文件或LUN将立即在活动卷中可用。

在对文件或LUN执行SnapRestore还原期间、不需要移动数据。但是、需要进行一些内部元数据更新、以反映文件或LUN中的底层块现在同时位于快照和活动卷中这一事实。此过程不会对性能产生任何影响、但会阻止创建快照、直到创建完成为止。根据所还原文件的总大小、处理速率约为5 Gbps (18 TB/小时)。

联机备份

在备份模式下保护和恢复Oracle数据库需要两组数据。请注意、这不是唯一的Oracle备份选项、但最常见。

- 备份模式下数据文件的快照
- 数据文件处于备份模式时创建的归档日志

如果需要完全恢复(包括所有已提交的事务)、则需要第三项：

- 一组当前的重做日志

可以通过多种方法恢复联机备份。许多客户使用ONTAP命令行界面还原快照、然后使用Oracle RMAN或sqlplus完成恢复。在大型生产环境中、这种情况尤为常见、在这些环境中、数据库还原的概率和频率极低、任何还原操

作步骤都由技能娴熟的数据库管理人员来处理。为了实现完全自动化、NetApp SnapCenter等解决方案包括一个具有命令行和图形界面的Oracle插件。

一些大型客户采用了一种更简单的方法、即在主机上配置基本脚本、以便在特定时间将数据库置于备份模式、以便为计划的快照做准备。例如、计划命令 `alter database begin backup` 23：58时、 `alter database end backup` 00：02、然后将快照直接计划在午夜在存储系统上。这样、便形成了一个简单、高度可扩展的备份策略、无需外部软件或许可证。

数据布局

最简单的布局是将数据文件隔离到一个或多个专用卷中。它们必须未受任何其他文件类型的污染。这是为了确保数据文件卷可以通过SnapRestore操作快速还原、而不会销毁重要的重做日志、控制文件或归档日志。

SAN对专用卷中的数据文件隔离具有类似要求。对于Microsoft Windows等操作系统、一个卷可能包含多个数据文件LUN、每个LUN都具有一个NTFS文件系统。对于其他操作系统、通常会有一个逻辑卷管理器。例如、对于Oracle ASM、最简单的选择是将ASM磁盘组的LUN限制为一个可作为一个单元进行备份和还原的卷。如果出于性能或容量管理原因需要更多卷、则在新卷上创建更多磁盘组可简化管理。

如果遵循这些准则、则可以直接在存储系统上计划快照、而无需执行一致性组快照。原因是Oracle备份不需要同时备份数据文件。联机备份操作步骤旨在使数据文件能够持续更新、因为它们会在数小时内缓慢流式传输到磁带。

如果使用分布在卷之间的ASM磁盘组、则会出现复杂情况。在这些情况下、必须执行cG-Snapshot、以确保ASM元数据在所有成分卷之间保持一致。

*注意：*验证ASM `spfile` 和 `passwd` 文件不在托管数据文件的磁盘组中。这会影响有选择地还原数据文件和仅还原数据文件的能力。

本地恢复过程**—NFS**

此操作步骤可以手动驱动、也可以通过SnapCenter等应用程序驱动。基本操作步骤如下所示：

1. 关闭数据库。
2. 将数据文件卷恢复到所需还原点之前的快照。
3. 将归档日志重放至所需位置。
4. 如果需要完全恢复、则重放当前重做日志。

此操作步骤假定所需的归档日志仍存在于活动文件系统中。否则、必须还原归档日志、或者可以将RMAN/sqlplus定向到快照目录中的数据。

此外、对于较小的数据库、最终用户可以直接从中恢复数据文件 `.snapshot` 目录、而无需自动化工具或存储管理员协助即可执行 `snaprestore` 命令：

本地恢复过程**—SAN**

此操作步骤可以手动驱动、也可以通过SnapCenter等应用程序驱动。基本操作步骤如下所示：

1. 关闭数据库。
2. 将托管数据文件的磁盘组静置。操作步骤因所选的逻辑卷管理器而异。使用ASM时、此过程需要卸载磁盘组。对于Linux、必须卸载文件系统、并且必须停用逻辑卷和卷组。目标是停止要还原的目标卷组上的所有更新。

3. 将数据文件磁盘组还原到所需还原点之前的快照。
4. 重新激活新还原的磁盘组。
5. 将归档日志重放至所需位置。
6. 如果需要完全恢复、请重放所有重做日志。

此操作步骤假定所需的归档日志仍存在于活动文件系统中。否则、必须通过使归档日志LUN脱机并执行还原来还原归档日志。这也是一个将归档日志划分为专用卷非常有用的示例。如果归档日志与重做日志共享一个卷组、则必须先将重做日志复制到其他位置、然后才能还原整个一组LUN。此步骤可防止丢失这些最终记录的事务。

Storage Snapshot优化的备份

在Oracle 12c发布后、基于Snapshot的备份和恢复变得更加简单、因为无需将数据库置于热备份模式。因此、可以直接在存储系统上计划基于快照的备份、同时仍保留执行完整或时间点恢复的能力。

尽管数据库管理器(操作步骤)对数据库管理器(数据库管理器)比较熟悉、但长期以来、可以使用数据库处于热备份模式时未创建的快照。在恢复期间、需要对Oracle 10g和11g执行额外的手动步骤、才能使数据库保持一致。采用Oracle 12c、 `sqlplus` 和 `rman` 包含额外的逻辑、用于重放未处于热备份模式的数据文件备份上的归档日志。

如前文所述、恢复基于快照的热备份需要两组数据：

- 在备份模式下创建的数据文件的快照
- 数据文件处于热备份模式时生成的归档日志

在恢复期间、数据库会从数据文件读取元数据、以选择恢复所需的归档日志。

经过存储快照优化的恢复需要略有不同的数据集才能实现相同的结果：

- 数据文件的快照、以及用于标识快照创建时间的方法
- 从最近的数据文件检查点到快照的确切时间的归档日志

在恢复期间、数据库会从数据文件中读取元数据、以确定所需的最早归档日志。可以执行完全恢复或时间点恢复。执行时间点恢复时、了解数据文件快照的时间至关重要。指定恢复点必须在快照创建时间之后。NetApp建议为快照时间至少添加几分钟、以考虑时钟变化。

有关完整的详细信息、请参见Oracle 12c文档各个版本中有关"使用存储Snapshot优化进行恢复"主题的Oracle文档。另请参见Oracle文档ID 604683.1、了解有关Oracle第三方快照支持的信息。

数据布局

最简单的布局是将数据文件隔离到一个或多个专用卷中。它们必须未受任何其他文件类型的污染。这是为了确保数据文件卷可以通过SnapRestore操作快速还原、而不会销毁重要的重做日志、控制文件或归档日志。

SAN对专用卷中的数据文件隔离具有类似要求。对于Microsoft Windows等操作系统、一个卷可能包含多个数据文件LUN、每个LUN都具有一个NTFS文件系统。对于其他操作系统、通常也会有一个逻辑卷管理器。例如、对于Oracle ASM、最简单的选择是将磁盘组限制为一个卷、该卷可以作为一个单元进行备份和还原。如果出于性能或容量管理原因需要更多卷、则在新卷上创建更多磁盘组可简化管理。

如果遵循这些准则、则可以直接在ONTAP上计划快照、而无需执行一致性组快照。原因是针对快照优化的备份不需要同时备份数据文件。

如果ASM磁盘组分布在多个卷中、则会出现复杂情况。在这些情况下、必须执行cG-Snapshot、以确保ASM元数据在所有成分卷之间保持一致。

[注]验证ASM spfile和passwd文件是否不在托管数据文件的磁盘组中。这会影响有选择地还原数据文件和仅还原数据文件的能力。

本地恢复过程**—NFS**

此操作步骤可以手动驱动、也可以通过SnapCenter等应用程序驱动。基本操作步骤如下所示：

1. 关闭数据库。
2. 将数据文件卷恢复到所需还原点之前的快照。
3. 将归档日志重放至所需位置。

此操作步骤假定所需的归档日志仍存在于活动文件系统中。否则、必须还原归档日志、或 `rman` 或 `sqlplus` 可以定向到中的数据 `.snapshot` 目录。

此外、对于较小的数据库、最终用户可以直接从中恢复数据文件 `.snapshot` 目录、而无需借助自动化工具或存储管理员来执行SnapRestore命令。

本地恢复过程**—SAN**

此操作步骤可以手动驱动、也可以通过SnapCenter等应用程序驱动。基本操作步骤如下所示：

1. 关闭数据库。
2. 将托管数据文件的磁盘组静置。操作步骤因所选的逻辑卷管理器而异。使用ASM时、此过程需要卸载磁盘组。对于Linux、必须卸载文件系统、并停用逻辑卷和卷组。目标是停止要还原的目标卷组上的所有更新。
3. 将数据文件磁盘组还原到所需还原点之前的快照。
4. 重新激活新还原的磁盘组。
5. 将归档日志重放至所需位置。

此操作步骤假定所需的归档日志仍存在于活动文件系统中。否则、必须通过使归档日志LUN脱机并执行还原来还原归档日志。这也是一个将归档日志划分为专用卷非常有用的示例。如果归档日志与重做日志共享一个卷组、则必须在还原整个LUN集之前将重做日志复制到其他位置、以避免丢失最终记录的事务。

完全恢复示例

假设数据文件已损坏或销毁、需要完全恢复。要执行此操作的操作步骤如下所示：

```
[oracle@host1 ~]$ sqlplus / as sysdba
Connected to an idle instance.
SQL> startup mount;
ORACLE instance started.
Total System Global Area 1610612736 bytes
Fixed Size                  2924928 bytes
Variable Size            1040191104 bytes
Database Buffers          553648128 bytes
Redo Buffers               13848576 bytes
Database mounted.
SQL> recover automatic;
Media recovery complete.
SQL> alter database open;
Database altered.
SQL>
```

时间点恢复示例

整个恢复操作步骤只需一个命令： `recover automatic`。

如果需要时间点恢复、则快照的时间戳必须已知、并且可按如下方式进行标识：

```
Cluster01::> snapshot show -vserver vserver1 -volume NTAP_oradata -fields
create-time
vserver   volume        snapshot   create-time
--------  ------------  ---------  ------------------------
vserver1  NTAP_oradata  my-backup  Thu Mar 09 10:10:06 2017
```

快照创建时间显示为3月9日和10：10：06。为了安全起见、快照时间增加了一分钟：

```
[oracle@host1 ~]$ sqlplus / as sysdba
Connected to an idle instance.
SQL> startup mount;
ORACLE instance started.
Total System Global Area 1610612736 bytes
Fixed Size                  2924928 bytes
Variable Size            1040191104 bytes
Database Buffers          553648128 bytes
Redo Buffers               13848576 bytes
Database mounted.
SQL> recover database until time '09-MAR-2017 10:44:15' snapshot time '09-
MAR-2017 10:11:00';
```

此时将启动恢复。考虑到可能的时钟差异、它指定了10：11：00的快照时间(比记录的时间晚一分钟)和10：44的目标恢复时间。接下来、sqlplus请求所需的归档日志、以达到所需的恢复时间10：44。

```
ORA-00279: change 551760 generated at 03/09/2017 05:06:07 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_31_930813377.dbf
ORA-00280: change 551760 for thread 1 is in sequence #31
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 552566 generated at 03/09/2017 05:08:09 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_32_930813377.dbf
ORA-00280: change 552566 for thread 1 is in sequence #32
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 553045 generated at 03/09/2017 05:10:12 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_33_930813377.dbf
ORA-00280: change 553045 for thread 1 is in sequence #33
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 753229 generated at 03/09/2017 05:15:58 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_34_930813377.dbf
ORA-00280: change 753229 for thread 1 is in sequence #34
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
Log applied.
Media recovery complete.
SQL> alter database open resetlogs;
Database altered.
SQL>
```

使用使用快照完成数据库恢复 `recover automatic` 命令不需要特定的许可、但需要使用进行时间点恢复 `snapshot time` 需要Oracle高级压缩许可证。

数据库管理和自动化工具

ONTAP在Oracle数据库环境中的主要价值来自核心ONTAP技术、例如即时Snapshot副本、简单的SnapMirror复制以及高效创建FlexClone卷。

在某些情况下、直接在ONTAP上配置这些核心功能即可满足要求、但更复杂的需求则需要一个业务流程层。

SnapCenter

SnapCenter是NetApp的旗舰级数据保护产品。从很低的层面来看、它在执行数据库备份的方式上与SnapManager产品类似、但它是从头开始构建的、用于在NetApp存储系统上提供单一管理平台来进行数据保护管理。

SnapCenter包括一些基本功能、例如基于快照的备份和还原、SnapMirror和SnapVault复制、以及大型企业大规

模运行所需的其他功能。这些高级功能包括扩展的基于角色的访问控制(Role-Based Access Control、RBAC)功能、可与第三方业务流程产品集成的REST API、对数据库主机上的SnapCenter插件进行无中断集中管理、以及专为云规模环境设计的用户界面。

REST

ONTAP还包含丰富的ESTful API集。这样、第三方供应商就可以创建与ONTAP深度集成的数据保护和其他管理应用程序。此外、希望创建自己的自动化工作流和实用程序的客户也可以轻松使用这种ESTful API。