技术报告

基于 NetApp 存储的 Oracle 数据库最佳实践

NetApp 公司 Jeffrey Steiner
2014 年 3 月| TR-3633

重要信息

请查阅互操作性表工具 (Interoperability Matrix Tool, IMT) 确定本报告中指定的环境、配置和版本是否支持您的环境。

目录

表格目录

1 简介

由于用户需求（包括数据库大小、性能需求和数据保护需求）多种多样，因此，明确用于在 NetApp® 存储上配置 Oracle® 数据库的最佳实践少之又少。据了解，如今在 NetApp 存储上部署的系统大小各异，从在 VMware® ESX® 环境下运行约 6,000 个数据库的虚拟化环境到当前容量为 810 TB 且在不断增长的单实例数据仓库，可谓多种多样。

本文档阐述了在 NetApp 存储上部署 Oracle 数据库的一部分实际要求，并根据 Oracle 存储解决方案架构师的特定业务需求探讨了必须考虑的诸多设计因素。文档中的主题先介绍适用于所有环境的一般考虑因素，然后再介绍专门针对网络文件系统 (Network File System, NFS) 和存储区域网络 (Storage Area Network, SAN) 的一般要求，最后，按字母顺序列出了适用于各个操作系统的特定建议。

2 一般 Data ONTAP 配置建议

本文不会全面阐述 NetApp Data ONTAP® 的各项配置。为具有 2,000 个虚拟化数据库的环境设计的最佳实践可能并不适用于配置有三个大型 ERP 数据库的环境；即便数据保护和恢复要求稍作改动，也可能会显著影响存储设计。本节探讨了一些基本细节，但要获得设计方面的全面协助，请联系 NetApp 或 NetApp 经销商。

2.1 基于 Snapshot 的备份

在文件系统布局方面，最重要的考虑因素是，对如何利用 Snapshot™ 备份进行规划。主要方法有两种：

- 崩溃状态一致的备份
- 由 Snapshot 保护的热备份

崩溃状态一致的数据库备份要求在单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。如果数据库存储在一个 FlexVol® 卷（灵活卷）中，则该过程会很简单：可以随时创建 Snapshot 副本。如果数据库跨越多个卷，则必须创建一致性组 (Consistency Group, CG) Snapshot 副本。可以借助多个选项来创建 CG Snapshot 副本，包括 Snap Creator™ 软件、SnapManager® for Oracle、SnapDrive® for UNIX® 以及用户维护的脚本。

在备份点恢复即可满足要求的情况下，主要使用崩溃状态一致的 Snapshot 备份。在某些情况下可以应用归档日志，但如果需要进行更精确的时间点恢复，则最好使用热备份。

基于 Snapshot 的热备份的基本步骤如下：

1. 将数据库置于热备份模式。
2. 为托管数据文件的所有卷创建 Snapshot 副本。
3. 退出热备份模式。
4. 执行 alter system archive log current 强制执行日志归档。
5. 为托管归档日志的所有卷创建 Snapshot 副本。

之后会生成一组 Snapshot 副本，这些副本包含：(a) 热备份模式下的数据文件 (b) 热备份模式下生成的关键归档日志。这两项数据是恢复数据库的两项必备条件。此外，为方便起见，还应对控制文件等其他文件进行保护，但只有数据文件和归档日志是必不可少的两个条件。

尽管不同客户可能会采用完全不同的策略，但这些策略几乎都会遵循本节前面所述的原则。

2.2 基于 Snapshot 的恢复

为 Oracle 数据库设计卷布局时，必须首先做出一项特定决策：是否使用基于卷的 SnapRestore® 技术。

利用基于卷的 SnapRestore (Volume-based SnapRestore, VBSR)，您几乎可以将卷状态即时还原到之前的时间点，但这也意味着将还原卷上的**所有**数据。而许多使用情形并不适用于这一点。例如，如果整个数据库（包括数据文件、重做日志和归档日志）存储在一个 FlexVol 卷上，并且此卷是使用 VBSR 恢复的，则数据可能会丢失，因为可能会丢弃经过更新的归档日志和重做数据。

恢复操作不需要 VBSR，使用基于文件的 SnapRestore (File-based SnapRestore, SFSR) 或只是将文件从 Snapshot 副本复制回活动文件系统中即可恢复许多数据库。

数据库非常庞大或必须尽快恢复时，首选 VBSR。此时需要隔离数据文件。在 NFS 环境中，给定数据库的数据文件必须存储在未受其他任何类型文件影响的专用 FlexVol 卷中。在 SAN 环境中，数据文件必须存储在专用 FlexVol 卷上的专用 LUN 中。如果使用卷管理器（包括 ASM），数据文件还必须具有专用的磁盘组。

以此方式隔离数据文件可以在不损坏其他文件系统的情况下将数据文件的状态还原为先前状态。

2.3 精简配置

在 Oracle 环境中，精简配置的使用会受到限制，因为 Oracle 会在创建数据文件时将其初始化为完整大小。对 Oracle 环境进行精简配置必须格外小心，因为数据变更率会意外提高。例如，如果为表重新编制索引，Snapshot 副本空间消耗则会快速增长，或者放错位置的 RMAN 备份会在非常短的时间内写入大量数据。最终，在数据文件扩展期间，如果文件系统用尽可用空间，将难以恢复 Oracle 数据库。

大多数问题可通过配置卷自动增长和 Snapshot 自动删除策略来避免。

NFS

在 NFS 环境中使用 Oracle 的大多数客户都会对其 Oracle 数据库进行配置，使其自动扩展数据文件，并使用卷自动增长来确保卷中具有足够的可用空间。

SAN

文件系统环境中的精简配置效率可能会逐渐下降，因为删除和擦除的数据在文件系统中占用的未分配空白空间会越来越多。

精简配置在逻辑卷层的使用效率更高。使用逻辑卷管理器（如 Veritas™ VxVM 或 Oracle ASM）时，底层 LUN 会划分为多个块区。这些块区仅在必要时使用。例如，如果数据库大小开始为 2 TB，而随着时间的推移可能会增长到 10 TB，则可以将其放置按 LVM 磁盘组组织的 10 TB 精简配置 LUN 上。数据库在创建时只占用 2 TB 的磁盘空间，只有为满足数据库增长而分配块区时才占用额外空间。

3 一般 Oracle 配置建议

以下参数通常适用于所有配置。

3.1 filesystemio_options

该 Oracle 初始化参数用于控制异步 I/O 和直接 I/O 的使用情况。与一般的认知相反，异步 I/O 和直接 I/O 并不互相排斥。NetApp 发现，该参数在客户环境中往往会配置不当。这种配置不当直接导致了许多性能问题。

从本质上说，异步 I/O 意味着可以并行处理 Oracle I/O 操作。用户需要配置大量 dbwriter 进程并更改服务器进程配置，之后才能在各种操作系统上使用异步 I/O。使用异步 I/O 后，操作系统自身便可代表数据库软件以并行方式高效执行 I/O。这不会为数据带来风险，而且关键操作（如 Oracle 重做日志记录）仍同步执行。

直接 I/O 意味着绕过操作系统缓冲区缓存。UNIX 系统上的 I/O 通常流经操作系统缓冲区缓存。这一点对于不保留内部缓存的应用程序来说十分重要，但 Oracle 在 SGA 中具有自己的缓冲区缓存。在几乎所有情况下，启用直接 I/O 并向 Oracle SGA 分配服务器 RAM 都要比依靠操作系统缓冲区缓存效果更好。Oracle SGA 使用内存的效率比较高。此外，当 I/O 流经操作系统缓冲区时，它会受到其他处理的影响，从而会使延迟有所增加。对于写入负载繁重的 I/O 来说，这一延迟增加会尤其显著，而这种 I/O 对低延迟的要求又比较严格。

适用于 `filesystemio_options` 的选项可以汇总如下：

- Asynchronous I/O（异步 I/O）。Oracle 应将 I/O 请求提交给操作系统进行处理。这样，Oracle 便可继续其他工作而不是等待 I/O 完成，而且可提高 I/O 的并行程度。
- Direct I/O（直接 I/O）。Oracle 应直接对物理文件执行 I/O，而不是通过主机操作系统缓存来路由 I/O。

- None（无）。使用同步 I/O 和缓冲 I/O。在这些配置中，共享和专用服务器进程之间的选择以及 dbwriter 的数量将变得更加重要。
- Setall。同时使用异步 I/O 和直接 I/O。

几乎在所有情况下，使用 setall 都是最佳选择，但应考虑以下因素：

- 一些客户过去可能已遇到异步 I/O 问题，尤其是在使用 Red Hat Enterprise Linux® (RHEL4) 版本时。系统将不再报告这些问题，异步 I/O 在所有当前操作系统上都很稳定。
- 如果数据库一直都在使用缓冲 I/O，则切换到直接 I/O 可能还要更改 SGA 大小。如果禁用缓冲 I/O，则会导致主机操作系统的数据库缓存性能优势丢失。而重新向 SGA 添加 RAM 则会再次获得这种优势。最终结果是，I/O 性能应有所提高。
- 尽管对 Oracle SGA 使用操作系统 RAM 几乎始终会优于使用主机操作系统缓冲区缓存，但有时却无法确定最佳值。例如，如果数据库服务器包含许多 Oracle 实例，而这些实例间歇性活动，则使用 SGA 非常小的缓冲 I/O 或许是最佳方案。这样，所有正在运行的数据库实例便可以灵活地使用操作系统上的剩余可用 RAM。此种情况极其少见，但已在某些客户站点出现过。

  注： `filesystemio_options` 参数在 DNFS 和 ASM 环境中无效。使用直接 NFS (Direct NFS, DNFS) 或自动存储管理 (Automatic Storage Management, ASM) 会导致同时自动使用异步 I/O 和直接 I/O。

3.2 db_file_multiblock_read_count

此参数控制 Oracle 在顺序 I/O 期间单次操作读取的 Oracle 数据库块的最大数量。

- 它并非 Oracle 在任何及所有读取操作期间读取块的数量，不影响随机 I/O，仅影响顺序 I/O。
- Oracle 建议用户不要设置此参数，数据库软件会自动设置最佳值。这通常意味着此参数将设置为生成 1 MB 大小的 I/O 值。例如，要读取 1 MB 的数据块，每个数据块为 8 k，则需要读取 128 个块，因此，该参数的默认值将为 128。
- NetApp 在客户站点发现的大多数数据库性能问题均与该参数设置不当有关。在 Oracle 8 和 Oracle 9 中，此值发生变化是有原因的；因为该数据库会原位升级到 Oracle 10 及更高版本，因此，该参数可能会自动保存在 `init.ora` 文件中而用户毫不知情。与默认值 128 相比，原有设置 8 或 16 会严重损害顺序 I/O 性能。

正如用户所见，没有证据证明更改为此值会为性能带来明显的优势，因此不应在 `init.ora` 文件中设置该参数。

3.3 重做块大小

Oracle 支持 512 字节或 4 k 字节的重做块大小。默认为 512 字节。最佳选项应为 512 字节，因为这样会最大程度地减少重做操作期间写入的数据量。但是，当日志记录速率非常高时，4 k 大小可能会在性能方面表现出优势。例如，重做块越大，重做日志记录速率为 50 MB/秒的单个数据库的效率可能就越高。如果存储系统支持许多重做日志记录总量非常大的数据库，则 4 k 的重做块会带来优势，因为它只需要更新 4 k 块的一部分，从而避免了效率低下的不完整 I/O 处理。

只有当 NetApp 或 Oracle 客户支持专门提出建议时，才应更改默认块大小，并且应根据对所运行的数据库的实际 I/O 模式进行分析的结果来进行更改。

4 Oracle RAC

本节适用于 Oracle 10.2.0.2 及更高版本。有关 Oracle 的早期版本，请参阅本文档以及 Oracle 文档 294430.1 以确定最佳设置。

4.1 disktimeout

主要的存储相关 Real Application Cluster (RAC) 参数为 `disktimeout`。该参数控制着表决文件 I/O 完成的阈值。如果超过 `disktimeout`，RAC 节点将被逐出并重新启动。

该参数的默认值为 200。对于标准集群接管/交还过程来说，该值应足以满足要求。NetApp 强烈建议在 RAC 投入生产之前对其配置进行全面测试，因为许多因素都会影响接管/交还。除了完成存储故障转移所需的时间之外，以下操作也需要额外的时间进行处理：传播链路聚合控制协议 (Link Aggregation Control Protocol, LACP) 更改；SAN 多路径软件检测 I/O 超时并在备用路径上重试；在处理表决磁盘 I/O 之前对大量 I/O 进行排队和重试（数据库异常活跃时）。

如果无法执行实际存储接管/交还，则可以通过对数据库服务器执行缆线拉拔测试来模拟效果。

4.2 misscount

`misscount` 参数通常仅影响 RAC 节点之间的网络检测信号。默认值通常为 30 秒。如果操作系统启动磁盘不在本地，则该参数可能就非常重要。其中包括启动磁盘位于通过 FC SAN 和 NFS 启动的操作系统上的主机，以及启动磁盘位于虚拟化数据存储库（如 VMDK 文件）上的主机。如果存储接管/交还中断了对启动磁盘的访问，则整个操作系统都可能会暂时挂起。Data ONTAP 完成存储操作以及操作系统更改路径并恢复 I/O 所需的时间可能会超过 `misscount` 阈值。这样，在恢复与启动 LUN 的连接之后，节点将立即逐出。大多数情况下，在发生逐出并随后进行重新启动时，不会记录任何日志消息来指示重新启动的原因。并非所有配置都会受影响，因此应测试 RAC 环境中的任何 SAN 启动主机、NFS 启动主机或基于数据存储库的主机，以便在与启动磁盘的通信中断时 RAC 仍保持稳定状态。

如果使用非本地启动磁盘，则可能需要更改 `misscount`，使其与 `disktimeout` 匹配。如果更改该参数，则应再次执行测试，以便同时了解对 RAC 行为（如节点故障转移时间）的影响。

5 在 Oracle 环境中使用 Flash Cache、Flash Pool 和 SSD

本文档并不详尽阐述对 Oracle 数据库使用闪存和 SSD 技术的情形，但某些常见问题和常见错误必须引起注意。

本节中介绍的所有原则对所有协议和文件系统（包括 Oracle ASM）均同等适用。

Flash Cache：flexscale.lopri_blocks

该参数适用于使用 Flash Cache™ 智能缓存的情况。此选项的默认值为 `off`，表示**不**应缓存与通常优先级比较低的块操作相关的 I/O（如随机覆盖和顺序 I/O）。原因很简单：大多数数据库都会受到随机读取操作延迟的限制。发生随机覆盖时，Oracle 数据库几乎总是会为该块保留一份副本，而且不太可能马上就进行重新读取，因此，缓存覆盖会浪费 Flash Cache 中的宝贵空间。当 Oracle 执行顺序读取 I/O 时，存储阵列本身就可以高效处理这种非常大的块操作，即使底层磁盘为 SATA 也是如此。Flash Cache 无益于此类 I/O，尝试缓存此类 I/O 通常会给 CPU 带来不必要的负载，并且同样会浪费 Flash Cache 中的宝贵空间；而如果将这些空间用于缓存随机 I/O，则效果可能会更好。

注：只有在详细咨询 NetApp 客户支持或专业服务后，才应更改此参数。

使用 SSD 聚合

将重做日志置于 SSD 聚合上是一种常见的错误。如果将 SSD 驱动器与直接连接的设备配合使用，则对提高日志记录性能极为有益，但 NetApp 阵列已经包含基于已镜像 NVRAM/NVMEM 的非易失性固态存储。Oracle 数据库执行写入操作时，一旦将此写入操作记录到 NVRAM/NVMEM 中，此写入操作就会获得确认。写入性能不受最终接收写入数据的驱动器的类型影响。

在最佳情况下，将 SSD 聚合用于托管顺序写入（如重做日志记录）或临时数据文件 I/O 不会有任何效果，但在多数情况下，SSD 聚合的设备数远远少于系统上的 SAS 或 SATA 聚合。NetApp 发现，如果将写入量庞大的顺序工作负载移动到设备过少的 SSD 聚合中，则会产生严重的性能问题。

SSD 聚合应预留用于涉及随机 I/O 的工作负载。索引尤其适合置于 SSD 驱动器上。NetApp 专业服务可协助分析 Oracle AWR 或 statspack 文件以进行更详细的分析。

Flash Pool：写入缓存

之前介绍的有关 Flash Cache 和 SSD 聚合的原则同样适用于 Flash Pool™ 智能缓存。将 Flash Pool 用于写入缓存更可能损害性能而非提高性能，原因如下：(a) 写入首先提交到 NVRAM/NVMEM 缓存；(b) Oracle 不太可能在写入后马上就重新读取，因此，这种低优先级写入 I/O 的缓存将取代更重要读取活动的缓存。但也存在例外情况 — 如果 Oracle 缓冲区缓存面临压力，而块因过期而退出缓存并等待不久被再次读取，此时例外情况尤其明显。如果 Oracle 块写入级别非常高，则建议对写入缓存的优势进行测试。

6 以太网配置

Oracle 数据库软件安装本身所需的 TCP/IP 设置通常足以为所有 NFS 或 iSCSI 存储资源提供良好性能。在某些使用情形中，NetApp 在 10GbE 环境中实施了网络适配器制造商的特定建议，显现出了性能优势。

6.1 以太网流量控制

此技术允许客户端请求发送方暂时停止数据传输。执行此操作通常是因为接收方无法尽快处理传入数据。与让接收方开始放弃数据包相比，以前请求发送方停止传输造成的中断更少，因为缓冲区已满。

近年来，因以太网流量控制所导致的性能问题日益增加。原因是，以太网流量控制在物理层进行。如果网络配置允许任何数据库服务器将以太网流量控制请求发送到存储系统，则结果就会造成**所有**已连接客户端中的 I/O 均暂停。随着一个存储控制器所处理的客户端数量不断增加，发送流量控制请求的可能性也随之增加。随着操作系统虚拟化的广泛应用，客户站点上的问题也会频繁出现。

NetApp 系统上的 NIC 不应接收流量控制请求。不同网络交换机制造商实现这一点的方法也有所不同。在大多数情况下，可以将流量控制设置为 `receive desired or receive on`，也就是说，流量控制请求不会转发给存储控制器。而在其他情况下，存储控制器上的网络连接可能不允许禁用流量控制。在这些情况下，必须通过更改数据库服务器自身的 NIC 配置或数据库服务器所连接的交换机端口来将客户端配置为始终不发送流量控制请求。

6.2 巨型帧

实践证明，使用巨型帧可以减少 CPU 和网络开销，从而可以在一定程度上提高 GbE 网络的性能，但所获的优势往往并不明显。即便如此，NetApp 仍然建议尽可能地尝试实施巨型帧，一方面可以尽可能地获得性能优势，一方面也让该解决方案适应未来需要。

应将在 10GbE 网络中使用巨型帧视为近乎强制性要求。原因是如果没有巨型帧，大多数 10GbE 实施都将在到达 10GbE 水平之前达到每秒数据包数限制。使用巨型帧可以提高 TCP/IP 的处理效率，原因是它允许数据库服务器、NIC 和存储系统处理的数据包数量更少但大小却更大。不同的 NIC 性能改善各不相同，但却非常显著。

一个常见的错误认知是，实施巨型帧需要所有连接设备都支持巨型帧。两个网络端点应协商可用性最高的帧，如建立连接时。在典型环境下，网络交换机的 MTU 大小设置为 9216、NetApp 控制器设置为 9000，客户端则是 9000 和 1514 的混合。可以支持 9000 MTU 的客户端将使用巨型帧，仅支持 1514 的客户端将协商较低的值。

在完全交换的环境中，很少出现问题。在路由环境中则必须格外谨慎，不要强制任何中间路由器将巨型帧分段。

6.3 TCP 参数

经常配置错误的设置有三个：TCP 时间戳、SACK 和 TCP 窗口缩放。许多过期文档继续存在于 Internet 上，因此建议禁用这些参数中的一个或多个以提高性能。这在多年前具有一些价值，那时，CPU 功能低很多，尽可能减少 TCP 处理开销会有所助益。

使用现代操作系统时，禁用上述任意 TCP 功能通常不会产生任何明显优势，也不会有损性能。而在虚拟化网络环境中，缺少这些功能则极易损害性能，因为高效处理数据包丢失和网络质量变动需要使用它们。应假设托管 Oracle 数据库的所有服务器均已启用 TCP 时间戳、SACK 和 TCP 窗口缩放。

7 一般 NFS 配置

7.1 安装和修补

ORACLE_HOME 中存在以下挂载选项将导致主机缓存被禁用：

```
cio, actimeo=0, noac, forcedirectio.
```

这会对 Oracle 软件的安装和修补速度产生严重的负面影响。许多客户会在安装或修补 Oracle 二进制文件期间暂时删除这些挂载选项。如果用户确认在安装或修补过程中无任何其他进程正在使用目标 ORACLE_HOME，则可以放心地完成此操作。

7.2 集群模式 Data ONTAP 和 NFS 流量控制

在某些情况下，使用集群模式 Data ONTAP 需要更改 Oracle 或 Linux 内核参数。原因与 NFS 流量控制有关。请不要将此与以太网流量控制相混淆。NFS 流量控制允许 NFS 服务器（如 Data ONTAP）限制与未确认接收数据的 NFS 客户端之间的网络通信。在出现故障的 NFS 客户端以超出其响应处理能力的速率请求数据时，这可以保护 NFS 服务器。如果没有保护，NFS 服务器上的网络缓冲区将填满未经确认的数据包。

在少数情况下，Oracle DNFS 客户端和较新的 Linux NFS 客户端中的阵发性 I/O 会超过集群模式 Data ONTAP NFS 服务器自我保护的极限。在继续发送请求获取更多数据的同时，NFS 客户端会滞后对入站数据的处理。这将导致 NFS 连接出现性能和稳定性问题。

尽管极少出现问题，但作为最佳实践，NetApp 建议采取下列保护措施。这仅适用于集群模式 Data ONTAP。这些更改不会对性能产生负面影响。

1. 使用 Oracle DNFS 时，请将 `DNFS_BATCH_SIZE` 参数设置为 `128`。Oracle 11.2.0.4 及更高版本中有此参数。对于较低版本的 Oracle，请联系 Oracle 客户支持了解是否有修补程序，或联系 NetApp 获取其他建议。
2. 使用较新的 Linux 版本时，请确保 TCP 插槽表有限。这些参数控制着可同时存在的未处理 NFS 操作的数量。运行 `sysctl -a` 可查看以下参数：

```
sunrpc.tcp_max_slot_table_entries
```

如果存在，请将下面的两个参数都设置为 `128`：

```
sunrpc.tcp_max_slot_table_entries
sunrpc.tcp_slot_table_entries.
```

更改这些参数后，将对同时存在的未处理 I/O 操作的数量进行限制。

7.3 NFS 锁定

如果 Oracle 数据库服务器崩溃，在重新启动后陈旧 NFS 锁定可能会出现问题。如果密切关注服务器上的名称解析配置，则可避免此问题。如果无法实现这一点，则需手动清除存储系统上的锁定。下面的文章详细解释了这些选项：https://kb.netapp.com/support/index?page=content&id=1010994。

陈旧锁定所产生的问题通常可以完全避免。NLM 锁定管理器使用 `uname -n` 确定主机名，而 rpc.statd 进程使用 `gethostbyname()` 确定主机名。这些均需与操作系统相匹配，才能正确清除陈旧锁定。例如，主机可能正在查找 filer5 所拥有的锁定，但这些锁定已被主机注册为 filer5.mydomain.org。如果 `gethostbyname()` 返回的值与 `uname -a` 不同，解锁过程将失败。

下面是用于验证名称解析是否完全一致的简单脚本：

```
#! /usr/bin/perl
$uname=`uname -n`;
chomp($uname);
($name, $aliases, $addrtype, $length, @addrs) = gethostbyname $uname;
print "uname -n yields: $uname\n";
print "print gethostbyname yields: $name\n";
```

如果 `gethostbyname` 与 `uname` 不匹配，则可能存在陈旧锁定。例如，此结果将显示潜在问题：

```
uname -n yields: filer5
print gethostbyname yields: filer5.mydomain.org
```

通常，可以通过更改 `/etc/hosts` 中的主机显示顺序来解决此问题。例如，假定主机文件包括以下条目：

```
10.156.110.201  filer5.mydomain.org filer5 loghost
```

解决方法是更改 FQDN 和短主机名的显示顺序。这将导致 `gethostbyname()` 返回短主机名 `filer5`。现在，主机名与 `uname` 输出相匹配，锁定将在服务器崩溃之后自动清除。

8 一般 SAN 配置

8.1 LUN 对齐

LUN 对齐是指对于底层文件系统布局优化 I/O。在 NetApp 系统上，存储块以 4 k 为单位。Oracle 数据文件上的一个 8 k 块应正好对应于两个 4 k 块。如果 LUN 配置中的错误使对齐在任一方向上移动 1 k，则每个 8 k Oracle 块都将存在于三个不同的 4 k 存储块上而不是只存在于两个块上。这将最终导致延迟增加，并在存储系统内执行更多 I/O。

通常，只有在未使用逻辑卷管理器时才关注 LUN 对齐，这意味着关注重点是使用 Linux 和 Solaris 时如何对齐 LUN。如果逻辑卷组内的物理卷是在整个磁盘设备上定义的（即未创建分区），则 LUN 上的第一个 4 k 块将与存储系统上的第一个 4 k 块对齐。这是正确的对齐。分区会产生问题，因为它们移动了操作系统使用 LUN 的开始位置。只要偏移量按整个 4 k 单元移动，LUN 即会对齐。在 Linux 环境中，逻辑卷组应构建在整个磁盘设备上。需要分区时，可以通过运行 fdisk –u 并验证每个分区的“起点”是否为 8 的倍数来检查对齐。

Solaris 环境更为复杂；有关进一步的信息，请参见相应的主机实用程序文档。

请参见 http://support.netapp.com/documentation/productlibrary/index.html?productID=61343。

注意

在 Solaris x86 环境中，必须格外小心才能正确对齐，因为大多数配置都有多个分区层。Solaris x86 分区片段通常存在于标准主启动记录分区表之上。

8.2 LUN 错位警告

Oracle 重做日志记录通常会产生错位的 I/O，而此 I/O 可能会导致生成关于 Data ONTAP 上的 LUN 错位的警告，这会让人产生误解。Oracle 重做日志记录会按顺序覆盖各种写入大小的重做日志文件。未与 4 k 边界对齐的日志写入操作通常不会引发性能问题，因为下一个日志写入操作将会补齐该块。最终结果是，Data ONTAP 几乎能够将所有写入都作为完整的 4 k 块进行处理，尽管某些 4 k 块中的数据会分别在两个操作中写入。

可使用 `sio` 或 `dd` 之类的实用程序验证对齐情况，这些实用程序可按定义的块大小生成 I/O，然后可使用 `stats` 命令查看存储系统上的 I/O 对齐统计信息。

8.3 LUN 计数

Oracle 数据库性能会受到通过 SCSI 层执行并行 I/O 的能力影响。因此，两个 LUN 提供的性能比一个 LUN 更高。提高并行性的最简单方法是使用逻辑卷管理器，如 Veritas VxVM、Linux LVM2 或 Oracle Automatic Storage Management (ASM)。一般来说，将 LUN 数量增加到 8 个以上为 NetApp 客户带来的优势微乎其微，尽管对随机 I/O 负载较重的全固态驱动器 (Solid-state Drive, SSD) 环境进行的测试表明：性能可持续提高，直到 LUN 数量增加到 64 个。一般建议是，构建卷组时使用能够均匀分布 I/O 的块区大小。例如，若 1 TB 卷组由 10 个 100 GB LUN 构成且一个块区的大小为 100MB，这总共会产生 10,000 个块区（每个 LUN 1,000 个块区）。此 1 TB 卷组上的数据库生成的 I/O 应在所有 10 个 LUN 中均匀分布。

通常不需要进行条带化。大多数数据库都会受到随机 I/O 性能（而非顺序性能）的限制。如果数据文件分布在大量块区之间，则会在大量块区之间随机分布大量随机 I/O。这意味着，卷组中的所有 LUN 都会均衡使用，单个 LUN 不会对性能施加限制。

8.4 数据文件块大小

一些操作系统可以选择文件系统块大小。对于支持数据文件的文件系统，块大小应为 4 k。有时也可能支持更大的值，但该值必须是 4 k 的倍数。如果数据文件放置在块大小为 512 字节的文件系统上，则文件可能会错位。LUN 和文件系统可能已根据 NetApp 建议正确对齐，但文件 I/O 本身可能错位。这会产生严重的性能问题。

8.5 重做块大小

支持重做日志的文件系统使用的块大小必须是重做块大小的倍数。这通常要求重做日志文件系统和重做日志本身都使用 512 字节的块大小。重做速率非常高时，4 k 块大小可能性能更佳，因为这样可以在数量更少、效率更高的操作中执行 I/O。如果重做速率大于 50 MB/秒，请考虑使用 4 k 块大小并进行测试。

当客户在块大小为 4 k 且许多事务都非常小的文件系统上使用块大小为 512 字节的重做日志时，数据库出现了一些问题。将多个 512 字节更改应用于单个 4 k 文件系统块所涉及的开销曾导致性能问题，这些问题已通过将文件系统更改为使用 512 字节的块大小得到了解决。

9 AIX

9.1 一般注意事项

挂载选项 `cio` 在 IBM AIX 环境中极其重要。它可以防止序列化写入 I/O 和文件系统锁定操作产生性能限制。这既适用于 NFS 文件系统，也适用于 SAN 文件系统。

并发 I/O

挂载选项 `cio` 可启用并发 I/O。要在 AIX 系统上获得最佳性能，需要使用并发 I/O。如果不使用并发 I/O，则性能可能会受到限制，因为 AIX 会随后执行序列化原子 I/O，从而产生巨大开销。

用于并发 I/O 的最佳方法是使用 `init.ora` 参数 `filesystemio_options=setall`。该参数可以使 Oracle 打开特定文件，并将其用于并发 I/O。使用 `cio` 作为挂载选项会强制使用并发 I/O，而这可能会产生负面影响。例如，强制使用并发 I/O 将对文件系统禁用预读，这可能会损害 Oracle 数据库软件外部的 I/O（如复制和磁带备份）的性能。此外，Oracle GoldenGate 和 SAP® BR*Tools 等产品不支持对某些 Oracle 版本使用 `cio` 挂载选项。

因此，NetApp 不建议在文件系统级别使用 `cio` 挂载选项。应使用 `filesystemio_options=setall` 来启用并发 I/O。

9.2 AIX NFSv3 挂载选项

应使用以下选项。

表 1) 单个实例。

文件类型	挂载选项
ADR_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536
控制文件、数据文件、重做日志	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,[cio?]
ORACLE_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,intr

表 2) Real Application Cluster。

文件类型	挂载选项
ADR_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536
控制文件、数据文件、重做日志	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,[cio?],nointr,noac
CRS/表决	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,cio,nointr,noac
专用 ORACLE_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536
共享 ORACLE_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr

单实例挂载选项与 RAC 挂载选项之间的主要区别是 RAC 挂载选项中增加了 `noac`。这具有禁用主机操作系统缓存的效果，从而使 RAC 集群中的所有实例都具有一致的数据状态视图。尽管使用 `cio` 挂载选项和 init.ora 参数 `filesystemio_options=setall` 具有同样的禁用主机缓存的效果，但仍需使用 `noac`。

共享 ORACLE_HOME 部署需要 `noac` 的原因是为了使 Oracle 密码文件和 spfile 之类的文件保持一致。如果 RAC 集群中的每个实例都具有专用 ORACLE_HOME，则不需要此参数。

一些客户已报告因 ADR_HOME 位置中的数据 I/O 量过多所致的性能问题。删除 `noac` 挂载选项将会发生主机操作系统缓存并降低存储 I/O 级别。

注： 请在咨询 NetApp 和 Oracle 支持后执行此步骤。

9.3 AIX jfs/jfs2 挂载选项

应使用以下选项。

表 3) 单个实例。

文件类型	挂载选项
ADR_HOME	defaults
控制文件、数据文件、重做日志	defaults,[cio?]
ORACLE_HOME	defaults

在任何环境（包括数据库）中使用 AIX hdisk 设备之前，都应检查 queue_depth 参数。这并非 HBA 队列深度，而是与各个 hdisk 设备的 SCSI 队列深度相关。根据 LUN 的配置方式，queue_depth 的值可能过低，无法提供良好性能。测试表明，最佳值为 32–64。

10 HP-UX

10.1 HP-UX NFSv3 挂载选项

应使用以下选项。

表 4) 单个实例。

文件类型	挂载选项
ADR_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,suid
控制文件、数据文件、重做日志	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,forcedirectio,nointr,suid
ORACLE_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,suid

表 5) Real Application Cluster。

文件类型	挂载选项
ADR_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,noac,suid
控制文件、数据文件、重做日志	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr,noac,forcedirectio,suid
CRS/表决	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr,noac,forcedirectio,suid
专用 ORACLE_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,suid
共享 ORACLE_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr,noac,suid

单实例挂载选项与 RAC 挂载选项之间的主要区别是 RAC 挂载选项中增加了 noac 和 forcedirectio。这具有禁用主机操作系统缓存的效果，从而使 RAC 集群中的所有实例都具有一致的数据状态视图。尽管使用 init.ora 参数 filesystemio_options=setall 与禁用主机缓存具有同样的效果，但仍需使用 noac 和 forcedirectio。

共享 ORACLE_HOME 部署需要 noac 的原因是为了使 Oracle 密码文件和 spfile 之类的文件保持一致。如果 RAC 集群中的每个实例都具有专用 ORACLE_HOME，则不需要此参数。

一些客户已报告因 ADR_HOME 位置中的数据 I/O 量过多所致的性能问题。删除 noac 挂载选项将会发生主机操作系统缓存并降低存储 I/O 级别。

注： 请在咨询 NetApp 和 Oracle 支持后执行此步骤。

10.2 HP-UX VxFS 挂载选项

托管 Oracle 二进制文件的文件系统应使用以下挂载选项：

```
delaylog,nodatainlog
```

在 HP-UX 版本不支持并发 I/O 的情况下，包含数据文件、重做日志、归档日志和控制文件的文件系统应使用以下挂载选项：

```
nodatainlog,mincache=direct,convosync=direct
```

在支持并发 I/O 的情况下（VxFS 5.0.1 及更高版本，或具有 ServiceGuard Storage Management Suite），包含数据文件、重做日志、归档日志和控制文件的文件系统应使用以下挂载选项：

```
delaylog,cio
```

注： 参数 db_file_multiblock_read_count 在 VxFS 环境中尤为重要。除非另有明确指示，否则 Oracle 建议在 Oracle 10*g*™ R1 及更高版本中将此参数保留为未设置状态。对于 Oracle 8 k 块大小，该参数的默认值为 128。如果将此参数值强制设置为 16 或更小，使用 convosync=direct mount 选项则会损害顺序 I/O 性能，因而应删除该选项。这会损害其他方面的性能，因此，仅在真正需要更改 db_file_multiblock_read_count 值的默认值时，才应采取此步骤。

11 Linux

11.1 一般注意事项

Linux 上的 NFS 性能主要取决于名为 tcp_slot_table_entries 的参数。此参数控制 Linux 操作系统上允许的未处理 NFS 操作的数量。

大多数 2.6 衍生内核（包括 RH5 和 OL5）中的默认值均为 16，由此频繁引发性能问题。然而，在取消了 tcp_slot_table_entries 值上限的较新内核中却出现相反问题，系统因收到过多请求而导致存储问题。

解决方法是将此值设置为固定值。将 NetApp NFS 存储与 Oracle 数据库配合使用的所有 Linux 操作系统应使用的值都是 128。

RHEL6.2 及早期版本

可通过在 /etc/sysctl.conf 中加入以下条目来设置此参数：

```
sunrpc.tcp_slot_table_entries = 128
```

此外，使用 2.6 内核时，大多数 Linux 版本中都存在错误。启动进程在加载 NFS 客户端之前读取 /etc/sysctl.conf 的内容，这样，NFS 客户端最终加载后便采用默认值 16。要避免此问题，请将 /etc/init.d/netfs 编辑为在第一行脚本中调用 /sbin/sysctl -p，使 NFS 在挂载任何文件系统之前将 tcp_slot_table_entries 设置为 128。

RHEL6.3 及更高版本

应在使用 RHEL 6.3 及更高版本的客户端的 RPC 文件中应用以下修改：

```
echo "options sunrpc udp_slot_table_entries=64 tcp_slot_table_entries=128
tcp_max_slot_table_entries=128" >> /etc/modprobe.d/sunrpc.conf
```

11.2 Linux NFSv3 挂载选项

应使用以下选项。

表 6) 单个实例。

文件类型	挂载选项
ADR_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536
控制文件、数据文件、重做日志	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr
ORACLE_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr

表 7) Real Application Cluster。

文件类型	挂载选项
ADR_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,actimeo=0
控制文件、数据文件、重做日志	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr, actimeo=0
CRS/表决	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr,noac, actimeo=0
专用 ORACLE_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536
共享 ORACLE_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr, actimeo=0

单实例挂载选项与 RAC 挂载选项之间的主要区别是 RAC 挂载选项中增加了 `actimeo=0`。这具有禁用主机操作系统缓存的效果，从而使 RAC 集群中的所有实例都具有一致的数据状态视图。尽管使用 `init.ora` 参数 `filesystemio_options=setall` 与禁用主机缓存具有同样的效果，但仍需使用 `actimeo=0`。

共享 ORACLE_HOME 部署需要 `actimeo=0` 的原因是为了使 Oracle 密码文件和 spfile 之类的文件保持一致。如果 RAC 集群中的每个实例都具有专用 ORACLE_HOME，则不需要此参数。

一些客户已报告因 ADR_HOME 位置中的数据 I/O 量过多所致的性能问题。删除 `actimeo=0` 挂载选项将会发生主机操作系统缓存并降低存储 I/O 级别。

注： 请在咨询 NetApp 和 Oracle 支持后执行此步骤。

Linux Direct NFS

启用 DNFS 且在嵌套挂载中在单个服务器上多次挂载源卷时，需要一个额外的参数。此情况主要发生在支持 SAP 应用程序的环境中。例如，NetApp 系统上的单个卷可以在 /vol/oracle/base 上有一个目录，在 /vol/oracle/home 上有另一个目录。如果 /vol/oracle/base 挂载在 /oracle 上、/vol/oracle/home 挂载在 /oracle/home 上，结果将产生来自同一个源的嵌套 NFS 挂载。

操作系统可以检测到 /oracle 和 /oracle/home 驻留在同一个卷中，也就是在同一个源文件系统中。然后使用相同的设备句柄访问数据。这可提高操作系统缓存和某些其他操作的使用效率，但却会妨碍 DNFS。如果 DNFS 需要访问 /oracle/home 上的文件（如 spfile），则它可能会错误地尝试使用错误的数据路径，导致读取或写入操作失败。在这些配置中，应将 `nosharecache` 挂载选项添加到与该主机上的其他 NFS 文件系统共享源 FlexVol 卷的 NFS 文件系统中。这样可以强制 Linux 操作系统为此文件系统分配单独的设备句柄。

11.3 Linux ext3/ext4 挂载选项

NetApp 建议使用默认挂载选项。

12 Solaris

12.1 Solaris NFSv3 挂载选项

应使用以下选项。

表 8) 单个实例。

文件类型	挂载选项
ADR_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536
控制文件、数据文件、重做日志	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr,llock,suid
ORACLE_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,suid

事实证明，使用 llock 可以避免在获取并释放存储系统上的锁定时产生延迟，从而显著提高客户环境的性能。如果对环境中的大量服务器进行配置，使其挂载相同的文件系统，或者在环境中对 Oracle 进行配置，使其挂载这些数据库，则应谨慎使用此选项。此配置很少见，但确实存在。如果某个实例再次意外启动，则数据可能会损坏，因为 Oracle 将无法检测外部服务器上的锁定文件。NFS 锁定不会在其他情况下提供保护；比如在版本 3 中，只是建议使用它们。

由于 `llock` 和 `forcedirectio` 参数互斥，因此，请务必在 init.ora 文件中加入 `filesystemio_options=setall`，以便可以使用 `directio`。如果没有此参数，则会使用主机操作系统缓冲区缓存，并且可能会对性能产生负面影响。

表 9) Real Application Cluster。

文件类型	挂载选项
ADR_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,noac
控制文件、数据文件、重做日志	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr,noac,forcedirectio
CRS/表决	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr,noac,forcedirectio
专用 ORACLE_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,suid
共享 ORACLE_HOME	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr,noac,suid

单实例挂载选项与 RAC 挂载选项之间的主要区别是 RAC 挂载选项中增加了 `noac` 和 `forcedirectio`。这具有禁用主机操作系统缓存的效果，从而使 RAC 集群中的所有实例都具有一致的数据状态视图。尽管使用 init.ora 参数 `filesystemio_options=setall` 与禁用主机缓存具有同样的效果，但仍需使用 `noac` 和 `forcedirectio`。

共享 ORACLE_HOME 部署需要 `actimeo=0` 的原因是为了使 Oracle 密码文件和 spfile 之类的文件保持一致。如果 RAC 集群中的每个实例都具有专用 ORACLE_HOME，则不需要此参数。

一些客户已报告因 ADR_HOME 位置中的数据 I/O 量过多所致的性能问题。删除 `noac` 挂载选项将会发生主机操作系统缓存并降低存储 I/O 级别。

注： 请在咨询 NetApp 和 Oracle 支持后执行此步骤。

12.2 Solaris UFS 挂载选项

NetApp 强烈建议使用 logging 挂载选项，这样，在 Solaris 主机崩溃或 FC 连接中断时才会保持数据完整性，并且 Snapshot 备份可供使用。

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。

NetApp 对本报告中提供的任何信息或建议的准确性、可靠性或适用性以及因采用在此提供的信息或建议而可能导致的任何后果不做任何声明或担保。本文档中信息按原意发布，对此类信息的使用或对此处任意建议或技术的实施均由客户承担责任，并取决于客户评估和将其融入客户运作环境的能力。本文档以及文档中所含信息仅可用于本文档中所讨论的 NetApp 产品。

Go further, faster®

TR-3633-0314-zhCN

NetApp
www.netapp.com/cn/