存储配置
Enterprise applications

NetApp
May 09, 2024

目录

存储配置

FC SAN

Oracle数据库I/O的LUN对齐

LUN对齐是指针对底层文件系统布局优化I/O。

在ONTAP系统上、存储以4 KB为单位进行组织。一个数据库或文件系统的8 KB块应正好映射到两个4 KB块。如果LUN配置错误使对齐在任一方向上移动1 KB、则每个8 KB块将位于三个不同的4 KB存储块上、而不是两个。这种安排会增加发生原因延迟、并在存储系统中执行发生原因额外的I/O。

对齐也会影响LVM架构。如果在整个驱动器设备上定义了逻辑卷组中的物理卷(不创建分区)、则LUN上的第一个4 KB块与存储系统上的第一个4 KB块对齐。这是正确的对齐方式。分区会出现问题、因为它们会移动操作系统使用LUN的起始位置。只要偏移量以4 KB的整数单位移动、LUN就会对齐。

在Linux环境中、在整个驱动器设备上构建逻辑卷组。如果需要分区、请运行以检查对齐情况 `fdisk -u` 并验证每个分区的起始位置是否为八的倍数。这意味着分区从八个512字节扇区的倍数开始、即4 KB。

另请参见一节中有关数据压缩块对齐的讨论 "效率"。与8 KB压缩块边界对齐的任何布局也与4 KB边界对齐。

未对齐警告

数据库重做/事务日志记录通常会生成未对齐的I/O、此I/O可能会导致发生原因发出有关ONTAP上LUN错位的警告、从而使人产生误解。

日志记录会使用不同大小的写入顺序写入日志文件。不与4 KB边界对齐的日志写入操作通常不会出现发生原因性能问题、因为下一个日志写入操作会完成块。因此、ONTAP几乎能够将所有写入作为完整的4 KB块进行处理、即使某些4 KB块中的数据是在两个单独的操作中写入的。

使用等实用程序验证对齐情况 `sio` 或 `dd` 可以按定义的块大小生成I/O。可以使用查看存储系统上的I/O对齐统计信息 `stats` 命令：请参见 "WAFL对齐验证" 有关详细信息 …

Solaris环境中的对齐更为复杂。请参见 "ONTAP SAN 主机配置" 有关详细信息 …

小心
在Solaris x86环境中，请格外注意正确对齐，因为大多数配置都有多个分区层。Solaris x86分区片通常位于标准主引导记录分区表之上。

Oracle数据库LUN大小调整和LUN计数

要获得Oracle数据库的最佳性能和易管理性、选择最佳LUN大小和要使用的LUN数量至关重要。

LUN是ONTAP上的一个虚拟化对象、位于托管聚合中的所有驱动器上。因此、LUN的性能不受其大小的影响、因为无论选择何种大小、LUN都会利用聚合的全部性能潜能。

为了方便起见、客户可能希望使用特定大小的LUN。例如、如果数据库是基于LVM或Oracle ASM磁盘组构建的、其中每个磁盘组包含两个1 TB的LUN、则该磁盘组必须以1 TB为增量进行增长。最好使用八个500 GB

的LUN来构建磁盘组、以便可以以较小的增量来增加磁盘组。

建议不要建立通用标准LUN大小、因为这样做会使易管理性复杂化。例如、如果数据库或数据存储库的大小介于1 TB到2 TB之间、则100 GB的标准LUN大小可能效果良好、但20 TB的数据库或数据存储库需要200个LUN。这意味着、服务器重新启动时间会更长、需要在各种用户界面中管理更多对象、SnapCenter等产品必须对许多对象执行发现。使用更少、更大的LUN可避免此类问题。

- LUN计数比LUN大小更重要。
- LUN大小主要由LUN计数要求控制。
- 避免创建超出所需数量的LUN。

LUN计数

与LUN大小不同、LUN计数会影响性能。应用程序性能通常取决于通过SCSI层执行并行I/O的能力。因此、两个LUN的性能优于一个LUN。使用Veritas VLVM、Linux LVM2或Oracle ASM等LVM是提高并行性的最简单方法。

虽然对随机I/O非常繁重的100% SSD环境进行的测试表明、LUN数量最多可增加到64个、但一般来说、NetApp客户从LUN数量增加到16个以上所获得的优势微乎其微。

- NetApp建议*：

通常、四到十六个LUN足以满足任何给定数据库工作负载的I/O需求。由于主机SCSI实施的限制、如果LUN数量少于四个、则可能会造成性能限制。

Oracle数据库LUN放置

数据库LUN在ONTAP卷中的最佳放置位置主要取决于各种ONTAP功能的使用方式。

Volumes

首次接触ONTAP的客户通常会感到困惑的一点是、FlexVol的使用、通常简称为"卷"。

卷不是LUN。这些术语与许多其他供应商产品(包括云提供商)同义。ONTAP卷只是管理容器。它们不会自行提供数据、也不会占用空间。它们是文件或LUN的容器、旨在提高和简化易管理性、尤其是大规模管理。

卷和LUN

相关LUN通常位于同一个卷中。例如、需要10个LUN的数据库通常会将所有10个LUN放置在同一个卷上。

- 采用1：1的LUN与卷比率(即每个卷一个LUN)是一种*不*正式的最佳实践。
- 而是应将卷视为工作负载或数据集的容器。每个卷可能有一个LUN、也可能有多个LUN。正确的问题解答取决于易管理性要求。
- 将LUN分散在不必要数量的卷上可能会导致额外开销和操作计划问题、例如快照操作、UI中显示的对象数量过多、并导致在达到LUN限制之前达到平台卷限制。

卷、LUN和快照

Snapshot策略和计划放置在卷上、而不是LUN上。如果包含10个LUN的数据集位于同一个卷中、则这些LUN只

需要一个Snapshot策略。

此外、在一个卷中将给定数据集的所有相关LUN同位可实现原子快照操作。例如、如果基础LUN都位于一个卷上、则驻留在10个LUN上的数据库或包含10个不同操作系统的基于VMware的应用程序环境可以作为一个一致的对象进行保护。如果将它们放置在不同的卷上、则快照可能会(也可能不会)完全同步、即使是同时计划的也是如此。

在某些情况下、由于恢复要求、可能需要将一组相关LUN拆分为两个不同的卷。例如、一个数据库可能有四个用于数据文件的LUN和两个用于日志的LUN。在这种情况下、最好使用包含4个LUN的数据文件卷和包含2个LUN的日志卷。原因是独立可恢复性。例如、可以有选择地将数据文件卷还原到先前的状态、这意味着所有四个LUN都将还原到快照的状态、而日志卷及其关键数据不会受到影响。

卷、**LUN**和**SnapMirror**

SnapMirror策略和操作与快照操作一样、在卷上执行、而不是在LUN上执行。

通过在一个卷中将相关LUN同位、您可以创建一个SnapMirror关系、并通过一次更新来更新所有包含的数据。与快照一样、更新也是一项原子操作。保证SnapMirror目标具有源LUN的单个时间点副本。如果LUN分布在多个卷上、则这些副本之间可能一致、也可能不一致。

卷、**LUN**和**QoS**

虽然可以有选择地将QoS应用于各个LUN、但在卷级别设置QoS通常更容易。例如、给定ESX服务器中子系统使用的所有LUN都可以放置在一个卷上、然后应用ONTAP自适应QoS策略。因此、会产生一个可自行扩展的每TB IOPS限制、用于对所有LUN执行适用场景操作。

同样、如果数据库需要10万次IOPS并占用10个LUN、则在单个卷上设置一个10万次IOPS限制比在每个LUN上设置10个单独的10万次IOPS限制更容易。

多卷布局

在某些情况下、在多个卷之间分布LUN可能会很有用。主要原因是控制器条带化。例如、一个HA存储系统可能托管一个数据库、其中需要每个控制器的全部处理和缓存潜力。在这种情况下、典型的设计是、将一半的LUN放置在控制器1上的一个卷中、而将另一半LUN放置在控制器2上的一个卷中。

同样、控制器条带化也可用于负载平衡。如果HA系统托管100个数据库、每个数据库包含10个LUN、则可以设计该系统、其中每个数据库在两个控制器中的每个控制器上都接收一个5 LUN卷。这样、在配置更多数据库时、可以保证每个控制器的负载对称。

但是、这些示例均不涉及卷与LUN的比例为1：1。我们的目标仍然是通过在卷中主机代管相关LUN来优化易管理性。

例如、LUN与卷的比例为1：1就意味着容器化、在容器化中、每个LUN可能真正代表一个工作负载、需要逐个进行管理。在这种情况下、1：1的比例可能是最佳的。

Oracle数据库**LUN**大小调整和基于**LVM**的大小调整

当基于SAN的文件系统达到其容量限制时、可通过两种方法增加可用空间：

- 增加LUN的大小
- 将LUN添加到现有卷组并增加包含的逻辑卷

虽然可以选择调整LUN大小来增加容量、但通常最好使用LVM、包括Oracle ASM。存在LVM的一个主要原因是避免调整LUN大小。通过LVM、多个LUN会绑定到一个虚拟存储池中。从该池中划分出来的逻辑卷由LVM管理、并且可以轻松调整大小。另一个优势是、通过在所有可用LUN之间分布给定逻辑卷、可以避免特定驱动器上出现热点。通常、可以通过使用卷管理器将逻辑卷的底层块区重新定位到新LUN来执行透明迁移。

使用**Oracle**数据库进行**LVM**条带化

LVM条带化是指在多个LUN之间分布数据。结果是、许多数据库的性能显著提高。

在闪存驱动器时代之前、条带化用于帮助克服旋转驱动器的性能限制。例如、如果操作系统需要执行1 MB的读取操作、则从单个驱动器读取1 MB的数据将需要大量的驱动器磁头查找和读取、因为1 MB的传输速度较慢。如果在8个LUN上对1 MB的数据进行条带化、则操作系统可以问题描述并行执行8个128 K读取操作、从而减少完成1 MB传输所需的时间。

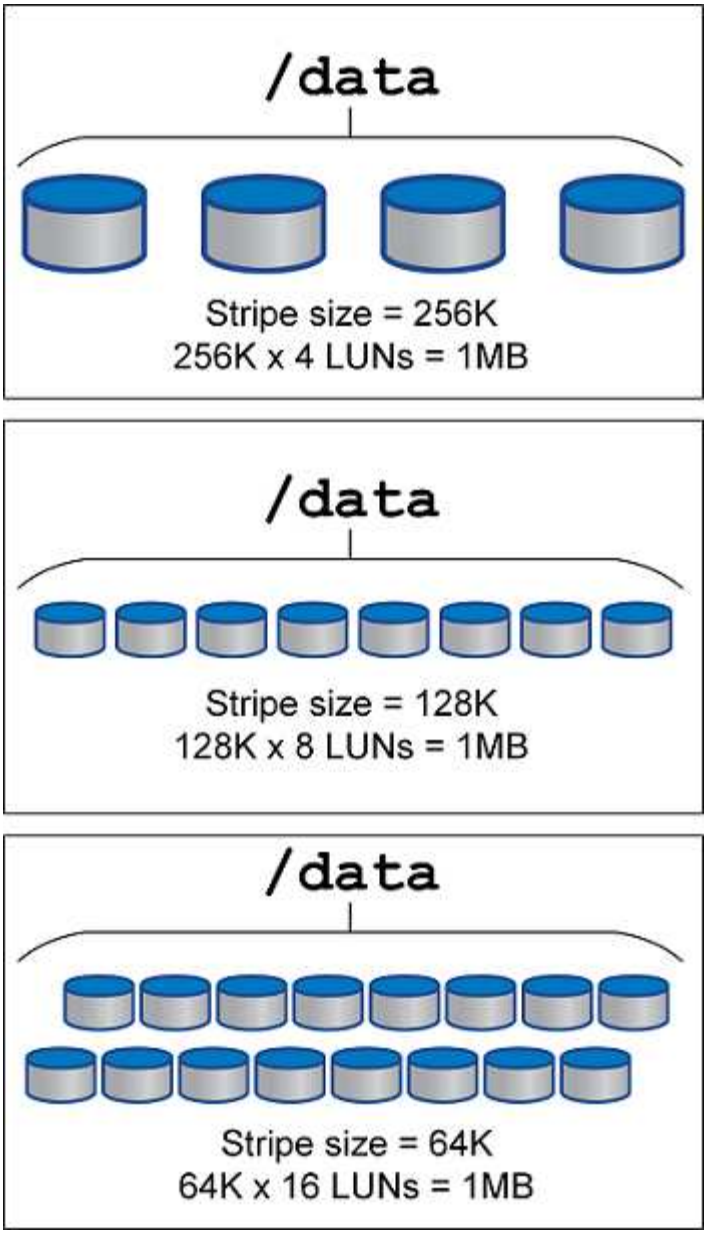
使用旋转驱动器进行条带化更为困难、因为必须事先知道I/O模式。如果条带化未正确调整为真正的I/O模式、则条带化配置可能会损害性能。使用Oracle数据库、尤其是使用全闪存配置时、条带化更易于配置、并且经验证可显著提高性能。

默认情况下、逻辑卷管理器(例如Oracle ASM)会进行条带化、但本机操作系统LVM则不会进行条带化。其中一些会将多个LUN绑定在一起、形成一个串联设备、从而导致数据文件只存在于一个LUN设备上。这会导致热点。其他LVM实施默认使用分布式块区。这与条带化类似、但更粗。卷组中的LUN会被划分为多个大块、称为块区、通常以MB为单位进行测量、然后逻辑卷会分布在这些块区中。结果是、文件的随机I/O应在各个LUN之间分布良好、但顺序I/O操作的效率不如所能达到的高。

性能密集型应用程序I/O几乎始终为(a)基本块大小单位或(b) 1兆字节。

条带化配置的主要目标是确保单文件I/O可作为一个单元执行、多块I/O (大小应为1 MB)可在条带化卷中的所有LUN之间均匀并行。这意味着条带大小不能小于数据库块大小、条带大小乘以LUN数量应为1 MB。

下图显示了三个可能的条带大小和宽度调整选项。选择LUN数量是为了满足上述性能要求、但在所有情况下、单个条带内的总数据均为1 MB。



NFS

适用于Oracle数据库的NFS配置

NetApp提供企业级NFS存储已超过30年、由于其精简性、随着向基于云的基础架构的推进、其使用量也在不断增长。

NFS协议包含多个版本、但要求各不相同。有关使用ONTAP的完整NFS配置问题描述、请参见 "TR-4067：《基于ONTAP的NFS最佳实践》"。以下各节介绍了一些更关键的要求和常见的用户错误。

NFS版本

NetApp必须支持操作系统NFS客户端。

- 遵循NFSv3标准的操作系统支持NFSv3。

- Oracle DNFS客户端支持NFSv3。
- 遵循NFSv4标准的所有操作系统均支持NFSv4。
- NFSv4.1和NFSv4.2需要特定的操作系统支持。请参见 "NetApp IMT" 支持的操作系统。
- 为NFSv4.1提供Oracle DNFS支持需要Oracle 12.2.0.2或更高版本。

。 "NetApp支持表" 对于NFSv3和NFSv4、不包括特定的操作系统。通常支持所有符合RFC的操作系统。在联机IMT中搜索NFSv3或NFSv4支持时、请勿选择特定操作系统、因为不会显示任何匹配项。常规策略隐式支持所有操作系统。

Linux NFSv3 TCP插槽表

TCP插槽表相当于主机总线适配器(Host Bus Adapter、HBA)队列深度的NFSv3。这些表可控制任何时候都可以处理的NFS操作的数量。默认值通常为16、该值太低、无法实现最佳性能。在较新的Linux内核上会出现相反的问题、这会自动将TCP插槽表限制增加到使NFS服务器充满请求的级别。

为了获得最佳性能并防止出现性能问题、请调整控制TCP插槽表的内核参数。

运行 `sysctl -a | grep tcp.*.slot_table` 命令、并观察以下参数：

```
# sysctl -a | grep tcp.*.slot_table
sunrpc.tcp_max_slot_table_entries = 128
sunrpc.tcp_slot_table_entries = 128
```

所有Linux系统都应包括 `sunrpc.tcp_slot_table_entries`，但只有部分包括 `sunrpc.tcp_max_slot_table_entries`。它们都应设置为128。

小心
如果未设置这些参数、可能会对性能产生显著影响。在某些情况下、性能会受到限制、因为Linux操作系统发出的I/O不足在其他情况下、随着Linux操作系统尝试问题描述的I/O数超过可处理的I/O数、I/O时间会增加。

ADr和NFS

一些客户报告了因中的数据量过多而导致的性能问题 `ADR` 位置。通常、只有在积累了大量性能数据之后、才会出现此问题。I/O过多的原因未知、但此问题似乎是由Oracle进程反复扫描目标目录以查找更改引起的。

卸下 `noac` 和 / 或 `actimeo=0` 挂载选项允许进行主机操作系统缓存并降低存储I/O级别。

* NetApp建议*不要放置 `ADR` 使用的文件系统上的数据 `noac` 或 `actimeo=0` 因为可能会出现性能问题。分开 `ADR` 如果需要、可将数据迁移到其他挂载点。

NFS-rootonly和mount-rootonly

ONTAP包含一个名为的NFS选项 `nfs-rootonly` 用于控制服务器是否接受来自高端口的NFS流量连接。作为一项安全措施、只有root用户才允许使用1024以下的源端口打开TCP/IP连接、因为此类端口通常保留供操作系统使用、而不是供用户进程使用。此限制有助于确保NFS流量来自实际操作系统NFS客户端、而不是模拟NFS客户端的恶意进程。Oracle DNFS客户端是用户空间驱动程序、但该进程以root用户身份运行、因此通常不需要更

改的值 `nfs-rootonly`。这些连接是从低端口进行的。

。 `mount-rootonly` 仅选件适用场景NFSv3。它控制是否从大于1024的端口接受RPC挂载调用。使用DNFS时、客户端将再次以root身份运行、因此它可以打开1024以下的端口。此参数无效。

通过NFS 4.0及更高版本打开与DNFS连接的进程不会以root身份运行、因此需要1024以上的端口。。 `nfs-rootonly` 必须将参数设置为disabled、DNFS才能完成连接。

条件 `nfs-rootonly` 处于启用状态、则会在打开DNFS连接的挂载阶段挂起。sqlplus输出类似于：

```
SQL>startup
ORACLE instance started.
Total System Global Area 4294963272 bytes
Fixed Size                  8904776 bytes
Variable Size             822083584 bytes
Database Buffers         3456106496 bytes
Redo Buffers                7868416 bytes
```

可以按如下方式更改此参数：

```
Cluster01::> nfs server modify -nfs-rootonly disabled
```

> 在极少数情况下、您可能需要将NFS-rootonly和mount-rootonly更改为disabled。如果服务器管理的TCP连接数量非常多、则可能没有低于1024的可用端口、并且操作系统会强制使用更高的端口。要完成连接、需要更改这两个ONTAP参数。

NFS导出策略：**super**用户 和**set_id**

如果Oracle二进制文件位于NFS共享上、则导出策略必须包括超级用户和set_id权限。

用于用户主目录等通用文件服务的共享NFS导出通常会强制转换root用户。这意味着挂载了文件系统的主机上的root用户发出的请求会重新映射为权限较低的其他用户。这有助于防止特定服务器上的root用户访问共享服务器上的数据、从而保护数据安全。在共享环境中、set_id位也可能存在安全风险。set_id位允许以与调用命令的用户不同的用户身份运行进程。例如、由root用户拥有且具有set_id位的shell脚本以root用户身份运行。如果其他用户可以更改该shell脚本、则任何非root用户都可以通过更新脚本以root用户身份问题描述命令。

Oracle二进制文件包含root用户拥有的文件、并使用set_id位。如果在NFS共享上安装了Oracle二进制文件、则导出策略必须包含适当的超级用户和set_id权限。在以下示例中、此规则同时包含这两者 `allow-suid` 和许可 `superuser` 使用系统身份验证的NFS客户端的(root)访问权限。

```
Cluster01::> export-policy rule show -vserver vserver1 -policyname orabin
-fields allow-suid,superuser
vserver   policyname ruleindex superuser allow-suid
--------- ---------- --------- --------- ----------
vserver1  orabin     1         sys       true
```

NFSv4/4.1配置

对于大多数应用程序、NFS3和NFSv4之间的差别非常小。应用程序I/O通常非常简单、不会从NFSv4中提供的某些高级功能中显著受益。从数据库存储角度来看、较高版本的NFS不应视为"升级"、而应视为包含其他功能的NFS版本。例如、如果需要Kerberos隐私模式(krb5p)的端到端安全性、则需要NFSv4。

*如果需要NFSv4功能、NetApp建议*使用NFSv4.1。在NFSv4.1中、NFSv4协议有一些功能增强功能、可提高某些边缘情况下的故障恢复能力。

与简单地将挂载选项从vs=3更改为vs=4.1相比、切换到NFSv4更为复杂。有关使用ONTAP配置NFSv4的更完整说明、包括有关配置操作系统的指导、请参见 "TR-4067：《基于ONTAP的NFS最佳实践》"。本技术报告的以下各节介绍了使用NFSv4的一些基本要求。

NFSv4域

有关NFSv4/4.1配置的完整说明不在本文档的讨论范围之内、但一个常见问题是域映射不匹配。从sysadmin的角度来看、NFS文件系统似乎运行正常、但应用程序会报告有关某些文件的权限和/或set_id的错误。在某些情况下、管理员错误地得出结论、认为应用程序二进制文件的权限已损坏、并在实际问题是域名时运行了chown或chmod命令。

在ONTAP SVM上设置NFSv4域名：

```
Cluster01::> nfs server show -fields v4-id-domain
vserver   v4-id-domain
--------- ------------
vserver1  my.lab
```

主机上的NFSv4域名在中进行设置 `/etc/idmap.cfg`

```
[root@host1 etc]# head /etc/idmapd.conf
[General]
#Verbosity = 0
# The following should be set to the local NFSv4 domain name
# The default is the host's DNS domain name.
Domain = my.lab
```

域名必须匹配。否则、中将显示类似以下内容的映射错误 `/var/log/messages`：

```
Apr 12 11:43:08 host1 nfsidmap[16298]: nss_getpwnam: name 'root@my.lab'
does not map into domain 'default.com'
```

应用程序二进制文件(如Oracle数据库二进制文件)包括root用户拥有的具有set_id位的文件、这意味着NFSv4域名不匹配会导致Oracle启动失败、并显示有关名为的文件的所有权或权限的警告 `oradism`，位于中 `$ORACLE_HOME/bin` 目录。它应如下所示：

```
[root@host1 etc]# ls -l /orabin/product/19.3.0.0/dbhome_1/bin/oradism
-rwsr-x--- 1 root oinstall 147848 Apr 17  2019
/orabin/product/19.3.0.0/dbhome_1/bin/oradism
```

如果此文件的所有权为mody、则可能存在NFSv4域映射问题。

```
[root@host1 bin]# ls -l oradism
-rwsr-x--- 1 nobody oinstall 147848 Apr 17  2019 oradism
```

要修复此问题、请选中 `/etc/idmap.cfg` 根据ONTAP上的v4-id-domain设置创建文件、并确保它们一致。如果不是、请进行所需的更改、然后运行 `nfsidmap -c`，然后等待片刻，让更改传播。然后、文件所有权应正确识别为root。如果用户尝试运行 `chown root` 更正NFS域配置之前、可能需要在此文件上运行 `chown root` 再次重申。

Oracle (Oracle)

Oracle数据库可以通过两种方式使用NFS。

首先、它可以使用通过操作系统中的本机NFS客户端挂载的文件系统。这有时称为内核NFS或kNFS。Oracle数据库挂载和使用NFS文件系统的方式与任何其他应用程序使用NFS文件系统的方式完全相同。

第二种方法是Oracle Direct NFS (DNFS)。这是在Oracle数据库软件中实施的NFS标准。它不会更改数据库管理程序配置或管理Oracle数据库的方式。只要存储系统本身具有正确的设置、DNFS的使用就应该对DBA团队和最终用户透明。

启用了DNFS功能的数据库仍会挂载常见的NFS文件系统。数据库打开后、Oracle数据库将打开一组TCP/IP会话并直接执行NFS操作。

直接**NFS**

Oracle的直接NFS的主要价值是绕过主机NFS客户端、直接在NFS服务器上执行NFS文件操作。要启用此功能、只需更改Oracle磁盘管理器(ODM)库即可。Oracle文档提供了此过程的说明。

使用DNFS可以显著提高I/O性能、并减少主机和存储系统上的负载、因为I/O是以尽可能最高效的方式执行的。

此外，Oracle DNFS还包括一个*选项*，用于实现网络接口多路径和容错。例如、可以将两个10 Gb接口绑定在一起、以提供20 Gb的带宽。一个接口发生故障会导致在另一个接口上重试I/O。整体操作与FC多路径非常相似。多路径早在几年前就已很常见、当时1 Gb以太网是最常用的标准。10 Gb NIC足以满足大多数Oracle工作负载的需求、但如果需要更多NIC、则可以绑定10 Gb NIC。

使用DNFS时、请务必安装Oracle文档1495104.1中所述的所有修补程序。如果无法安装修补程序、则必须对环境进行评估、以确保该文档中所述的错误不会出现发生原因问题。在某些情况下、无法安装所需的修补程序会导致无法使用DNFS。

请勿将DNFS与任何类型的轮叫名称解析结合使用、包括DNS、DDNS、NIS或任何其他方法。其中包括ONTAP中提供的DNS负载平衡功能。当使用DNFS的Oracle数据库将主机名解析为IP地址时、它在后续查找中不得更改。这可能会导致Oracle数据库崩溃并可能导致数据损坏。

对于依赖主机上挂载的可见文件系统的应用程序或用户活动、使用DNFS有时可能会出现发生原因问题、因为DNFS客户端会从主机操作系统带外访问文件系统。DNFS客户端可以在不了解操作系统的情况下创建、删除和修改文件。

如果使用单实例数据库的挂载选项、则可以缓存文件和目录属性、这也意味着可以缓存目录的内容。因此、DNFS可以创建文件、在操作系统重新读取目录内容和文件对用户可见之前、存在一个短暂的延迟。这通常不是问题，但在极少数情况下，SAP BR*Tools等实用程序可能会出现问题。如果发生这种情况、请更改挂载选项以使用针对Oracle RAC的建议来解决此问题。此更改会导致禁用所有主机缓存。

只有在以下情况下才更改挂载选项：(a)使用DNFS；(b)问题是由于文件可见性滞后而导致的。如果未使用DNFS、则在单实例数据库上使用Oracle RAC挂载选项会导致性能下降。

请参见有关的注释 `nosharecache` 在中 "Linux NFS挂载选项" 适用于可能会产生异常结果的Linux专用DNFS问题描述。

Oracle数据库和NFS租用和锁定

NFSv3处于无状态。这实际上意味着、NFS服务器(ONTAP)不会跟踪挂载了哪些文件系统、由谁挂载或哪些锁定真正到位。

ONTAP确实具有一些记录挂载尝试的功能、因此您可以了解哪些客户端可能正在访问数据、并且可能存在建议锁定、但该信息并不能保证100%完整。此操作无法完成、因为跟踪NFS客户端状态不是NFSv3标准的一部分。

NFSv4状态

相反、NFSv4是有状态的。NFSv4服务器可跟踪哪些客户端正在使用哪些文件系统、哪些文件存在、哪些文件和/或文件区域被锁定等 这意味着NFSv4服务器之间需要定期进行通信、以使状态数据保持最新。

NFS服务器所管理的最重要状态是NFSv4锁定和NFSv4租约、它们彼此交织在一起。您需要了解每种方法本身的工作原理、以及它们之间的关系。

NFSv4锁定

对于NFSv3、建议使用锁定。NFS客户端仍可修改或删除"锁定"文件。NFSv3锁定本身不会过期、必须将其删除。这会造成问题。例如、如果您有一个集群应用程序创建了NFSv3锁定、而其中一个节点发生故障、您该怎么办？您可以对运行正常的节点上的应用程序进行编码、以解除锁定、但您如何知道这是安全的？可能是"故障"节点正常运行、但未与集群的其余部分通信？

对于NFSv4、锁定的持续时间有限。只要持有锁定的客户端继续向NFSv4服务器签入、就不允许任何其他客户端获取这些锁定。如果客户端无法签入NFSv4、则锁定最终会被服务器撤消、其他客户端将能够请求并获取锁定。

NFSv4租约

NFSv4锁定与NFSv4租约关联。当NFSv4客户端与NFSv4服务器建立连接时、它将获得租约。如果客户端获得锁定(锁定类型有多种)、则锁定与租约关联。

此租约已定义超时。默认情况下、ONTAP会将超时值设置为30秒：

```
Cluster01::*> nfs server show -vserver vserver1 -fields v4-lease-seconds

vserver   v4-lease-seconds
--------- ----------------
vserver1  30
```

这意味着、NFSv4客户端需要每30秒与NFSv4服务器签入一次、才能续订其租约。

任何活动都会自动续订租约、因此、如果客户端正在执行工作、则无需执行添加操作。如果某个应用程序变得安静并且没有执行实际工作、则需要执行某种保活操作(称为序列)。基本上只是说"我还在这里、请刷新我的租约"。

Question: What happens if you lose network connectivity for 31 seconds? NFSv3处于无状态。它不需要来自客户端的通信。NFSv4是有状态的、租赁期过后、租约将过期、锁定将被撤消、锁定的文件将提供给其他客户端使用。

借助NFSv3、您可以四处移动网络缆线、重新启动网络交换机、更改配置、并确保不会发生任何不良事件。应用程序通常只需耐心等待网络连接重新工作即可。

使用NFSv4时、您有30秒的时间(除非您已在ONTAP中增加了该参数的值)来完成工作。如果超过此限制、您的租约将超时。通常、这会导致应用程序崩溃。

例如、如果您有一个Oracle数据库、并且网络连接丢失(有时称为"网络分区")、超过租约超时时间、则数据库将崩溃。

下面是一个示例、说明在发生这种情况时Oracle警报日志中会发生什么情况：

```
2022-10-11T15:52:55.206231-04:00
Errors in file /orabin/diag/rdbms/ntap/NTAP/trace/NTAP_ckpt_25444.trc:
ORA-00202: control file: '/redo0/NTAP/ctrl/control01.ctl'
ORA-27072: File I/O error
Linux-x86_64 Error: 5: Input/output error
Additional information: 4
Additional information: 1
Additional information: 4294967295
2022-10-11T15:52:59.842508-04:00
Errors in file /orabin/diag/rdbms/ntap/NTAP/trace/NTAP_ckpt_25444.trc:
ORA-00206: error in writing (block 3, # blocks 1) of control file
ORA-00202: control file: '/redo1/NTAP/ctrl/control02.ctl'
ORA-27061: waiting for async I/Os failed
```

如果您查看系统日志、应会看到以下几个错误：

```
Oct 11 15:52:55 host1 kernel: NFS: nfs4_reclaim_open_state: Lock reclaim
failed!
Oct 11 15:52:55 host1 kernel: NFS: nfs4_reclaim_open_state: Lock reclaim
failed!
Oct 11 15:52:55 host1 kernel: NFS: nfs4_reclaim_open_state: Lock reclaim
failed!
```

日志消息通常是问题的第一个迹象、而不是应用程序冻结。通常、网络中断期间不会显示任何内容、因为尝试访问NFS文件系统的进程和操作系统本身会被阻止。

网络重新正常运行后、将显示这些错误。在上面的示例中、重新建立连接后、操作系统尝试重新获取锁定、但太晚了。租约已过期、锁定已被删除。这会导致错误传播到Oracle层、并在警报日志中显示此消息。根据数据库的版本和配置、这些模式可能会有所不同。

总之、NFSv3可以承受网络中断、但NFSv4更敏感、并会规定一个明确的租赁期限。

如果30秒超时不可接受、该怎么办？如果您管理的网络动态变化、交换机重新启动或缆线重新定位会导致网络偶尔中断、该怎么办？您可以选择延长租赁期限、但是否要延长、需要说明NFSv4宽限期。

NFSv4宽限期

如果重新启动NFSv3服务器、它几乎可以立即提供IO。它并没有保持任何关于客户的状态。这样、ONTAP接管操作通常看起来接近瞬时。一旦控制器准备好开始提供数据、它就会向网络发送一个ARP、以指示拓扑发生变化。客户端通常会近乎即时地检测到这一点、数据将恢复流动。

但是、NFSv4会短暂暂停。这只是NFSv4工作原理的一部分。

NFSv4服务器需要跟踪租约、锁定以及谁在使用哪些数据。如果NFS服务器发生故障并重新启动、断电片刻或在维护活动期间重新启动、则会导致租用/锁定以及其他客户端信息丢失。在恢复操作之前、服务器需要确定哪个客户端正在使用哪些数据。这就是宽限期的所在。

如果您突然关闭并重新启动NFSv4服务器。恢复后、尝试恢复IO的客户端将收到一个响应、该响应本质上说："我丢失了租用/锁定信息。是否要重新注册您的锁？" 这是宽限期的开始。在ONTAP上、默认为45秒：

```
Cluster01::> nfs server show -vserver vserver1 -fields v4-grace-seconds

vserver   v4-grace-seconds
--------- ----------------
vserver1  45
```

因此、在重新启动后、控制器将暂停IO、而所有客户端都将回收其租约和锁定。宽限期结束后、服务器将恢复IO操作。

租赁超时与宽限期

宽限期和租赁期是连接的。如上所述、默认租约超时为30秒、这意味着NFSv4客户端必须至少每30秒向服务器签入一次、否则它们将失去租约、进而失去锁定。有一个宽限期、允许NFS服务器重建租用/锁定数据、默认为45秒。ONTAP要求宽限期比租赁期长15秒。这样可以确保设计为至少每30秒续订一次租约的NFS客户端环境

能够在重新启动后与服务器签入。45秒的宽限期可确保所有希望至少每30秒续订一次租约的客户都有机会续订租约。

如果不接受30秒的超时时间、您可以选择延长租赁期限。如果要将租约超时时间增加到60秒、以承受60秒网络中断、则必须将宽限期至少增加到75秒。ONTAP要求该期限比租赁期高15秒。这意味着、在控制器故障转移期间、IO暂停时间将更长。

这通常不会是问题。通常、用户每年只更新ONTAP控制器一次或两次、并且很少会因硬件故障而发生计划外故障转移。此外、如果您的网络可能会发生60秒的网络中断、并且您需要将租赁超时时间设置为60秒、则可能不会反对偶尔发生的存储系统故障转移、从而导致75秒的暂停。您已确认您的网络经常暂停60秒以上。

Oracle数据库的NFS缓存

如果存在以下任一挂载选项、则会禁用主机缓存：

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

这些设置可能会对软件安装、修补和备份/还原操作的速度产生严重的负面影响。在某些情况下、尤其是对于集群应用程序、由于需要在集群中的所有节点之间实现缓存一致性、因此必然需要使用这些选项。在其他情况下、客户会错误地使用这些参数、从而导致不必要的性能损害。

许多客户会在安装或修补应用程序二进制文件期间临时删除这些挂载选项。如果用户验证在安装或修补过程中没有其他进程正在使用目标目录、则可以安全地执行此删除。

Oracle数据库的NFS传输大小

默认情况下、ONTAP会将NFS I/O大小限制为64K。

大多数应用程序和数据库的随机I/O使用的块大小要小得多、远远低于64K的最大值。大型块I/O通常会并行处理、因此最大64K也不会限制获得最大带宽。

在某些工作负载中、最大64K会产生限制。特别是、如果数据库执行的I/O数量较少但规模较大、则单线程操作(例如备份或恢复操作或数据库完整表扫描)运行速度会更快、效率也会更高。ONTAP的最佳I/O处理大小为256K。

给定ONTAP SVM的最大传输大小可按如下方式进行更改：

```
Cluster01::> set advanced
Warning: These advanced commands are potentially dangerous; use them only
when directed to do so by NetApp personnel.
Do you want to continue? {y|n}: y
Cluster01::*> nfs server modify -vserver vserver1 -tcp-max-xfer-size
262144
Cluster01::*>
```

小心
请勿将ONTAP上允许的最大传输大小减小到低于当前挂载的NFS文件系统的rsize/wsize值。在某些操作系统中、这可能会导致挂起甚至数据损坏。例如、如果NFS客户端当前设置为rsize/wsize 65536,则ONTAP最大传输大小可以在65536- 1048576之间进行调整,但不会产生任何影响,因为客户端本身是有限的。将最大传输大小减小至65536,可能会损坏可用性或数据。

Oracle数据库和NVFAIL

NVFAIL是ONTAP中的一项功能、可确保在灾难性故障转移情形下的完整性。

数据库在存储故障转移事件期间容易损坏、因为它们会维护大量内部缓存。如果在发生灾难性事件时需要强制执行ONTAP故障转移或强制执行MetroCluster切换、而不管整体配置的运行状况如何、则先前确认的结果可能会被有效丢弃。存储阵列的内容会及时向后跳转、数据库缓存的状态不再反映磁盘上数据的状态。此不一致性会导致数据损坏。

缓存可以在应用程序层或服务器层进行。例如、如果Oracle Real Application Cluster (RAC)配置中的服务器在主站点和远程站点上都处于活动状态、则该配置会在Oracle SGA中缓存数据。如果强制切换操作导致数据丢失、则会使数据库面临损坏的风险、因为存储在SGA中的块可能与磁盘上的块不匹配。

在操作系统文件系统层使用缓存不太明显。装载的NFS文件系统中的块可能会缓存在操作系统中。或者、可以将基于主站点上LUN的集群文件系统挂载到远程站点的服务器上、然后再次缓存数据。在这些情况下、NVRAM故障、强制接管或强制切换可能会导致文件系统损坏。

ONTAP通过NVFAIL及其关联设置、保护数据库和操作系统免受这种情况的影响。

ASM Recandation Utility和ONTAP零块检测

启用实时压缩后、ONTAP可以高效删除写入文件或LUN的置零块。Oracle ASM Recasation Utility (ARU)等实用程序的工作方式是向未使用的ASM块区写入零。

这样、数据库管理器便可在删除数据后回收存储阵列上的空间。ONTAP会截获零并取消分配LUN中的空间。回收过程速度极快、因为存储系统中不会写入任何数据。

从数据库角度来看、ASM磁盘组包含零、读取这些LUN区域会产生零流、但ONTAP不会将零存储在驱动器上。而是进行简单的元数据更改、以便在内部将LUN的置零区域标记为任何数据为空。

出于类似的原因、涉及置零数据的性能测试无效、因为零块实际上不会在存储阵列中作为写入进行处理。

使用ARU时、请确保已安装Oracle建议的所有修补程序。