主机配置
Enterprise applications

NetApp
May 09, 2024

目录

主机配置

使用IBM AIX的Oracle数据库

使用ONTAP的IBM AIX上的Oracle数据库的配置主题。

并发I/O

要在IBM AIX上实现最佳性能、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会执行序列化的原子I/O、从而产生大量开销。

最初、NetApp建议使用 `cio` 挂载选项、用于强制在文件系统上使用并发I/O、但此过程存在缺点、不再需要。自AIX 5.2和Oracle 10gR1推出以来、AIX上的Oracle可以打开单个文件以实现并发IO、而不是强制在整个文件系统上执行并发I/O。

启用并发I/O的最佳方法是设置 `init.ora` 参数 `filesystemio_options` to `setall`。这样、Oracle就可以打开特定文件、以便用于并发I/O

使用 `cio` 作为挂载选项、会强制使用并发I/O、这可能会产生负面影响。例如、强制执行并发I/O会在文件系统上禁用预读、这可能会损害Oracle数据库软件之外发生的I/O的性能、例如复制文件和执行磁带备份。此外，Oracle GoldenGate和SAP BR*Tools等产品与不兼容 `cio` 适用于某些Oracle版本的挂载选项。

- NetApp建议*：
- 请勿使用 `cio` 文件系统级别的挂载选项。而是通过使用来启用并发I/O `filesystemio_options=setall`。
- 仅使用 `cio` 如果无法设置、则应设置挂载选项 `filesystemio_options=setall`。

AIX NFS挂载选项

下表列出了Oracle单实例数据库的AIX NFS挂载选项。

文件类型	挂载选项
ADr主页	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144`
控制文件 数据文件 重做日志	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144`
`ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,intr`

下表列出了RAC的AIX NFS挂载选项。

文件类型	挂载选项
ADr主页	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144`

文件类型	挂载选项
控制文件 数据文件 重做日志	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,noac`
`CRS/Voting`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,noac`
专用 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144`
共享 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr`

单实例挂载选项与RAC挂载选项之间的主要区别在于添加了 `noac` 挂载选项。这种添加的效果是禁用主机操作系统缓存、从而使RAC集群中的所有实例都能获得一致的数据状态视图。

但使用 `cio` 挂载选项和 `init.ora` 参数 `filesystemio_options=setall` 与禁用主机缓存具有相同的效果、但仍需要使用 `noac`。 `noac` 对于共享为必填项 `ORACLE_HOME` 部署以提高Oracle密码文件和等文件的一致性 `spfile` 参数文件。RAC集群中的每个实例都有一个专用 `ORACLE_HOME`，则不需要此参数。

AIX jfs/JFS2挂载选项

下表列出了AIX jfs/jfs2挂载选项。

文件类型	挂载选项
ADr主页	默认值
控制文件 数据文件 重做日志	默认值
ORACLE_HOME	默认值

使用AIX之前 `hdisk` 在任何环境(包括数据库)中、设备都应检查参数 `queue_depth`。此参数不是HBA队列深度、而是与各个的SCSI队列深度相关 `hdisk device. Depending on how the LUNs are configured, the value for `queue_depth` 可能太低、无法获得良好性能。测试表明、最佳值为64。

使用HP-UX的Oracle数据库

使用ONTAP在HP-UX上配置Oracle数据库主题。

HP-UX NFS挂载选项

下表列出了单个实例的HP-UX NFS挂载选项。

文件类型	挂载选项
ADr主页	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,suid`

文件类型	挂载选项
控制文件 数据文件 重做日志	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144,forc edirectio, nointr,suid`
`ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, suid`

下表列出了RAC的HP-UX NFS挂载选项。

文件类型	挂载选项
ADr主页	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, noac,suid`
控制文件 数据文件 重做日志	`rw, bg,hard, [vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,forcedirectio,suid`
CRS/表决	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac, forcedirectio,suid`
专用 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, suid`
共享 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,suid`

单实例挂载选项与RAC挂载选项之间的主要区别在于添加了 `noac` 和 `forcedirectio` 挂载选项。这种添加的效果是禁用主机操作系统缓存、从而使RAC集群中的所有实例都能获得一致的数据状态视图。但使用 `init.ora` 参数 `filesystemio_options=setall` 与禁用主机缓存具有相同的效果、但仍需要使用 `noac` 和 `forcedirectio`。

原因 `noac` 对于共享为必填项 `ORACLE_HOME` 部署是为了提高Oracle密码文件和spfile等文件的一致性。RAC集群中的每个实例都有一个专用 `ORACLE_HOME`，则不需要此参数。

HP-UX VxFS挂载选项

对于托管Oracle二进制文件的文件系统、请使用以下挂载选项：

```
delaylog,nodatainlog
```

对于包含数据文件、重做日志、归档日志和控制文件的文件系统、如果HP-UX版本不支持并发I/O、请使用以下挂载选项：

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

如果支持并发I/O (VxFS 5.0.1及更高版本或ServiceGuard Storage Management Suite)、请对包含数据文件、重做日志、归档日志和控制文件的文件系统使用以下挂载选项：

```
delaylog,cio
```

参数 `db_file_multiblock_read_count` 在VxFS环境中尤其重要。Oracle建议在Oracle 10g R1及更高版本中保持此参数未设置、除非另有明确指示。Oracle 8 KB块大小的默认值为128。如果此参数的值强制设置为16或更低、请删除 `convosync=direct` 挂载选项、因为它可能会损坏顺序I/O性能。此步骤会损害性能的其他方面、仅当的值为时才应执行此步骤 `db_file_multiblock_read_count` 必须更改默认值。

使用Linux的Oracle数据库

特定于Linux操作系统的配置主题。

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。

Linux NFS挂载选项

下表列出了单个实例的Linux NFS挂载选项。

文件类型	挂载选项
ADr主页	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144`
控制文件 数据文件 重做日志	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr`
ORACLE_HOME	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr`

下表列出了RAC的Linux NFS挂载选项。

文件类型	挂载选项
ADr主页	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,actimeo=0`
控制文件 数据文件 重做日志	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,actimeo=0`
CRS/表决	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,noac,actimeo=0`
专用 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144`
共享 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,actimeo=0`

单实例挂载选项与RAC挂载选项之间的主要区别在于添加了 `actimeo=0` 挂载选项。这种添加的效果是禁用主机操作系统缓存、从而使RAC集群中的所有实例都能获得一致的数据状态视图。但使用 `init.ora` 参数 `filesystemio_options=setall` 与禁用主机缓存具有相同的效果、但仍需要使用 `actimeo=0`。

原因 `actimeo=0` 对于共享为必填项 `ORACLE_HOME` 部署是为了提高Oracle密码文件和spfile等文件的一致性。RAC集群中的每个实例都有一个专用 `ORACLE_HOME`，则不需要此参数。

通常、非数据库文件应使用与单实例数据文件相同的选项进行挂载、但特定应用程序可能具有不同的要求。避免使用挂载选项 `noac` 和 `actimeo=0` 如果可能、因为这些选项会禁用文件系统级预读和缓冲。这可能会发生原因为提取、转换和加载等过程带来严重的性能问题。

access和**getattr**

一些客户注意到、极高级别的其他IOPS (如访问和getATTR)可能会主导其工作负载。在极端情况下、读取和写入等操作可能只占总数的10%。这是包含使用的任何数据库的正常行为 `actimeo=0` 和 / 或 `noac` 在Linux上、因为这些选项会对Linux操作系统执行发生原因操作、以便不断地从存储系统中重新加载文件元数据。访问和getattr等操作是低影响操作、可通过数据库环境中的ONTAP缓存进行处理。不应将其视为真正的IOPS、例如读取和写入、因为它们会对存储系统产生真正的需求。但是、其他这些IOPS确实会产生一些负载、尤其是在RAC环境中。要解决这种情况、请启用DNFS、它会绕过操作系统缓冲区缓存并避免执行这些不必要的元数据

操作。

Linux Direct NFS

一个额外的挂载选项、称为 `nosharecache`如果(a)启用了DNFS、并且(b)在单个服务器上多次挂载源卷(c)并使用嵌套NFS挂载、则需要使用此选项。此配置主要出现在支持SAP应用程序的环境中。例如、NetApp系统上的单个卷的目录可能位于 `/vol/oracle/base 然后、再按 /vol/oracle/home。条件 /vol/oracle/base 挂载于 /oracle 和 /vol/oracle/home 挂载于 `/oracle/home`的结果是来自同一源的嵌套NFS挂载。

操作系统可以检测到这一事实 /oracle 和 /oracle/home 位于同一个卷上、即同一个源文件系统。然后、操作系统会使用相同的设备句柄来访问数据。这样做可以改进操作系统缓存和某些其他操作的使用、但会干扰DNFS。如果DNFS必须访问某个文件、例如 spfile，开 /oracle/home，则可能会错误地尝试使用错误的数据路径。结果是I/O操作失败。在这些配置中、添加 nosharecache 将选项挂载到与该主机上的另一个NFS文件系统共享源FlexVol卷的任何NFS文件系统。这样做会强制Linux操作系统为该文件系统分配独立的设备句柄。

Linux Direct NFS和Oracle RAC

使用DNFS对于Linux操作系统上的Oracle RAC具有特殊的性能优势、因为Linux没有强制执行直接I/O的方法、而RAC需要执行直接I/O才能在节点间保持一致。作为临时解决策、Linux需要使用 actimeo=0 挂载选项、此选项会导致操作系统缓存中的文件数据立即过期。而此选项又会强制Linux NFS客户端不断重新读取属性数据、从而会损害延迟并增加存储控制器上的负载。

启用DNFS会绕过主机NFS客户端并避免这种损坏。多家客户报告说、在启用DNFS时、RAC集群的性能显著提高、ONTAP负载显著降低(尤其是与其他IOPS相关的负载)。

Linux Direct NFS和oranfstab文件

如果在Linux上使用DNFS和多路径选项、则必须使用多个子网。在其他操作系统上、可以使用建立多个DNFS通道 LOCAL 和 DONTROUTE 用于在一个子网上配置多个DNFS通道的选项。但是、这在Linux上不能正常工作、可能会导致意外的性能问题。在Linux中、用于DNFS流量的每个NIC都必须位于不同的子网上。

I/O计划程序

Linux内核允许对块设备的I/O计划方式进行低级控制。各种Linux发行版的默认值差别很大。测试表明、截止日期通常会获得最佳结果、但NOOP有时会稍好一些。性能差异极小、但如果需要从数据库配置中提取尽可能高的性能、请同时测试这两个选项。CFQ是许多配置中的默认设置、它已证明数据库工作负载存在严重的性能问题。

有关配置I/O计划程序的说明、请参见相关的Linux供应商文档。

多路径

某些客户在网络中断期间遇到崩溃、因为多路径守护进程未在其系统上运行。在最新版本的Linux上、操作系统和多路径守护进程的安装过程可能会使这些操作系统容易受到此问题的影响。软件包安装正确、但未配置为在重新启动后自动启动。

例如、RHEL5.5上的多路径守护进程的默认设置可能如下所示：

```
[root@host1 iscsi]# chkconfig --list | grep multipath
multipathd      0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

可使用以下命令更正此问题：

```
[root@host1 iscsi]# chkconfig multipathd on
[root@host1 iscsi]# chkconfig --list | grep multipath
multipathd      0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

ASM镜像

ASM 镜像可能需要更改 Linux 多路径设置，以使 ASM 能够识别问题并切换到备用故障组。ONTAP 上的大多数 ASM 配置都使用外部冗余，这意味着数据保护由外部阵列提供，并且 ASM 不会镜像数据。某些站点使用正常冗余的 ASM 来提供双向镜像，通常在不同站点之间进行镜像。

中显示的Linux设置 "NetApp主机实用程序文档" 包括导致I/O无限期排队的多路径参数这意味着、没有活动路径的LUN设备上的I/O会根据需要等待I/O完成。这通常是可取的、因为Linux主机会根据需要等待很长时间、以便SAN路径更改完成、FC交换机重新启动或存储系统完成故障转移。

这种无限制排队行为会导致ASM镜像出现问题、因为ASM必须收到I/O故障、才能在备用LUN上重试I/O。

在Linux中设置以下参数 `multipath.conf` 用于ASM镜像的ASM LUN文件：

```
polling_interval 5
no_path_retry 24
```

这些设置会为ASM设备创建120秒超时。超时计算为 `polling_interval` * `no_path_retry` 以秒为单位。在某些情况下、可能需要调整确切的值、但120秒的超时时间对于大多数使用来说应该足以满足要求。具体来说、120秒应允许控制器接管或恢复发生、而不会产生会导致故障组脱机的I/O错误。

a较低 `no_path_retry` 值可以缩短ASM切换到备用故障组所需的时间、但这也会增加在控制器接管等维护活动期间发生不必要故障转移的风险。可以通过仔细监控ASM镜像状态来缓解此风险。如果发生不必要的故障转移、并且重新同步执行速度相对较快、则可以快速重新同步镜像。对于追加信息、请参见有关使用的Oracle软件版本的ASM快速镜像重新同步的Oracle文档。

Linux xfs、ext3和ext4挂载选项

* NetApp建议*使用默认挂载选项。

使用ASMLi/AFD的Oracle数据库(ASM筛选器驱动程序)

特定于使用AFC和ASMlib的Linux操作系统的配置主题

ASMlib块大小

ASMlib是一个可选的ASM管理库和关联实用程序。其主要价值是能够使用可读标签将LUN或基于NFS的文件标记为ASM资源。

最新版本的ASMlib会检测到一个名为逻辑块/物理块指数(Logical Blocks Per Physical Block Exponent

、LBPPBE)的LUN参数。直到最近、ONTAP SCSI目标才报告此值。现在、它将返回一个值、指示首选的块大小为4 KB。这不是块大小的定义、但对于使用LBPPBE的任何应用程序来说、这是一个提示、即可以更高效地处理特定大小的I/O。但是、ASMlib会将LBPPBE解释为块大小、并在创建ASM设备时持久标记ASM标头。

此过程可能会在许多方面出现发生原因升级和迁移问题、所有这些问题都是由于无法在同一个ASM磁盘组中混用块大小不同的ASMlib设备。

例如、较早的阵列通常报告LBPPBE值为0、或者根本不报告此值。ASMlib会将此数据块大小解释为512字节。较新的阵列会被解释为块大小为4 KB。不能在同一个ASM磁盘组中混用512字节和4 KB设备。这样做会阻止用户使用两个阵列中的LUN或将ASM用作迁移工具来增加ASM磁盘组的大小。在其他情况下、RMAN可能不允许在块大小为512字节的ASM磁盘组与块大小为4 KB的ASM磁盘组之间复制文件。

首选解决方案是修补ASMlib。Oracle错误ID为13999609、此修补程序位于oracleasm-support-2.1.8-1及更高版本中。此修补程序允许用户设置参数 `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` to `true` 在中 `/etc/sysconfig/oracleasm` 配置文件。这样做会阻止ASMlib使用LBPPBE参数、这意味着新阵列上的LUN现在可识别为512字节块设备。

> 选项不会更改先前由ASMlib标记的LUN上的块大小。例如、如果必须将包含512字节块的ASM磁盘组迁移到报告4 KB块的新存储系统、则可以选择此选项 `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` 必须在新LUN标记为ASMlib之前进行设置。 如果设备已标记为oracleasm，则必须先重新格式化，然后再重新添加新的块大小。首先、使用取消配置设备 `oracleasm deletedisk`，然后使用清除设备的第一个1GB `dd if=/dev/zero of=/dev/mapper/device bs=1048576 count=1024`。最后、如果设备之前已分区、请使用 `kpartx` 命令删除陈旧分区或仅重新启动操作系统。

如果无法修补ASMlib、则可以从配置中删除ASMlib。此更改会造成系统中断、需要取消ASM磁盘的附加服务、并确保 `asm_diskstring` 参数设置正确。但是、这种更改不需要迁移数据。

ASM筛选驱动器((AFD)块大小

AAutomatic是一个可选的ASM管理库、它将取代ASMlib。从存储角度来看、它与ASMlib非常相似、但它还包括一些其他功能、例如、可以阻止非Oracle I/O、从而降低用户或应用程序错误可能损坏数据的几率。

设备块大小

与ASMlib一样、ADAD还会读取LUN参数Logical Blocks Per Physical Block Exponent (LBPPBE)、默认情况下会使用物理块大小、而不是逻辑块大小。

如果在现有配置中添加了AAutomatic、而ASM设备已格式化为512字节块设备、则可能会出现问题。AAutomatic驱动程序会将LUN识别为4K设备、如果ASM标签与物理设备不匹配、则会阻止访问。同样、迁移也会受到影响、因为不能在同一个ASM磁盘组中混用512字节和4 KB设备。这样做会阻止用户使用两个阵列中的LUN或将ASM用作迁移工具来增加ASM磁盘组的大小。在其他情况下、RMAN可能不允许在块大小为512字节的ASM磁盘组与块大小为4 KB的ASM磁盘组之间复制文件。

解决方案非常简单—AFAS包含一个参数、用于控制它使用的是逻辑块大小还是物理块大小。此全局参数会影响系统上的所有设备。要强制AfD使用逻辑块大小、请设置 `options oracleafd oracleafd_use_logical_block_size=1` 在中 `/etc/modprobe.d/oracleafd.conf` 文件

多路径传输大小

最近的Linux内核更改会强制实施发送到多路径设备的I/O大小限制、而AFD不会遵守这些限制。然后、I/O将被拒绝、从而导致LUN路径脱机。因此、无法安装Oracle Grid、配置ASM或创建数据库。

解决方案将在多路径.conf文件中为ONTAP LUN手动指定最大传输长度：

```
devices {
            device {
                vendor "NETAPP"
                product "LUN.*"
                max_sectors_kb 4096
        }
    }
```

即使当前不存在任何问题、如果使用AWAD来确保未来的Linux升级不会出现意外的发生原因问题、则应设置此参数。

使用Microsoft Windows的Oracle数据库

使用ONTAP在Microsoft Windows上配置Oracle数据库主题。

NFS

Oracle支持将Microsoft Windows与Direct NFS客户端结合使用。此功能提供了一条实现NFS管理优势的途径、其中包括跨环境查看文件、动态调整卷大小以及利用成本较低的IP协议。有关使用DNFS在Microsoft Windows上安装和配置数据库的信息、请参见Oracle官方文档。没有特别的最佳做法。

SAN

为了获得最佳压缩效率、请确保NTFS文件系统使用8K或更大的分配单元。使用4K分配单元(通常为默认分配单元)会对压缩效率产生负面影响。

Oracle数据库与Solaris

特定于Solaris OS的配置主题。

Solaris NFS挂载选项

下表列出了单个实例的Solaris NFS挂载选项。

文件类型	挂载选项
ADr主页	`rw,bg,hard,[vers=3,vers=4.1], roto=tcp, timeo=600,rsize=262144,wsize=262144`
控制文件 数据文件 重做日志	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,llock,suid`

文件类型	挂载选项
ORACLE_HOME	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,suid

的使用 llock 经验证、通过消除与获取和释放存储系统锁定相关的延迟、可以显著提高客户环境的性能。在配置了大量服务器以挂载相同文件系统且Oracle配置为挂载这些数据库的环境中、请谨慎使用此选项。尽管这种配置非常少见、但也有少数客户使用。如果某个实例再次意外启动、则可能会发生数据损坏、因为Oracle无法检测到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；与NFS版本3一样、它们仅为建议使用。

因为 llock 和 forcedirectio 参数是互斥的、这一点很重要 filesystemio_options=setall 中存在 init.ora 文件 directio 已使用。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不利影响。

下表列出了Solaris NFS RAC挂载选项。

文件类型	挂载选项
ADr主页	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,noac
控制文件 数据文件 重做日志	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,noac,forcedirectio
CRS/表决	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,noac,forcedirectio
专用 ORACLE_HOME	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,suid
共享 ORACLE_HOME	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,noac,suid

单实例挂载选项与RAC挂载选项之间的主要区别在于添加了 noac 和 forcedirectio 挂载选项。这种添加的效果是禁用主机操作系统缓存、从而使RAC集群中的所有实例都能获得一致的数据状态视图。但使用 init.ora 参数 filesystemio_options=setall 与禁用主机缓存具有相同的效果、但仍需要使用 noac 和 forcedirectio。

原因 actimeo=0 对于共享为必填项 ORACLE_HOME 部署是为了提高Oracle密码文件和spfile等文件的一致性。RAC集群中的每个实例都有一个专用 ORACLE_HOME，则不需要此参数。

Solaris UFS挂载选项

NetApp强烈建议使用日志记录挂载选项、以便在Solaris主机崩溃或FC连接中断时保持数据完整性。日志记录挂载选项还会保留Snapshot备份的可用性。

Solaris ZFS

必须仔细安装和配置Solaris ZFS，才能提供最佳性能。

mvector

Solaris 11对其处理大型I/O操作的方式进行了更改、这可能会导致SAN存储阵列出现严重的性能问题。NetApp错误报告“Solaris 11 ZFS性能回归”详细介绍了该问题。"解决方案将更改名为的操作系统参数 `zfs_mvector_max_size`。

以root用户身份运行以下命令：

```
[root@host1 ~]# echo "zfs_mvector_max_size/W 0t131072" |mdb -kw
```

如果此更改出现任何意外问题、可以通过以root用户身份运行以下命令轻松反转此更改：

```
[root@host1 ~]# echo "zfs_mvector_max_size/W 0t1048576" |mdb -kw
```

内核

要获得可靠的ZFS性能、需要对Solaris内核进行修补、以防止出现LUN对齐问题。此修复程序是在Solaris 10的修补程序147440-19以及Solaris 11的SRU 10.5中引入的。请仅将Solaris 10及更高版本与ZFS结合使用。

LUN配置

要配置LUN、请完成以下步骤：

1. 创建类型为的LUN `solaris`。
2. 安装指定的相应Host Utility Kit (HUK) "NetApp 互操作性表工具（ IMT ）"。
3. 完全按照所述执行HUK中的说明。基本步骤概述如下、但请参见 "最新文档" 正确的操作步骤。
   a. 运行 `host_config` 实用程序以更新 `sd.conf/sdd.conf` 文件这样可以使SCSI驱动器正确发现ONTAP LUN。
   b. 按照提供的说明进行操作 `host_config` 用于启用多路径输入/输出(MPIO)的实用程序。
   c. 重新启动。要在整个系统中识别任何更改、必须执行此步骤。
4. 对LUN进行分区并验证它们是否已正确对齐。有关如何直接测试和确认对齐的说明，请参阅“附录B：WAFL对齐验证”。

zpools

只能在中的步骤之后创建zpool "LUN配置" 执行。如果操作步骤未正确执行、则可能会因I/O对齐而导致性能严重下降。要在ONTAP上获得最佳性能、需要将I/O与驱动器上的4K边界对齐。在zpool上创建的文件系统使用有效块大小、该大小通过名为的参数进行控制 `ashift`，可通过运行命令来查看 `zdb -C`。

的值 `ashift` 默认为9、表示2^9或512字节。为了获得最佳性能、 `ashift` 值必须为12 (2^12=4k)。此值在创建zpool时设置、并且无法更改、这意味着具有的zpool中的数据 `ashift` 应通过将数据复制到新创建的zpool来

迁移12以外的文件。

创建zpool后、请验证的值 `ashift` 然后继续。如果此值不是12、则表示未正确发现LUN。销毁zpool、确认相关Host Utilities文档中显示的所有步骤均已正确执行、然后重新创建zpool。

zpool和Solaris LDom

Solaris LDOM还要求确保I/O对齐正确。虽然LUN可能会作为4K设备正确地被发现、但LDOM上的虚拟vdsk设备不会继承I/O域中的配置。基于该LUN的vdsk默认为512字节的块。

需要一个额外的配置文件。首先、必须为各个LLOM修补Oracle错误27824910、以启用其他配置选项。此修补程序已移植到所有当前使用的Solaris版本中。对LDOM进行修补后、即可按如下所示配置正确对齐的新LUN：

1. 确定要在新zpool中使用的一个或多个LUN。在此示例中、它是C2D1设备。

```
[root@LDOM1 ~]# echo | format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c2d0 <Unknown-Unknown-0001-100.00GB>
     /virtual-devices@100/channel-devices@200/disk@0
  1. c2d1 <SUN-ZFS Storage 7330-1.0 cyl 1623 alt 2 hd 254 sec 254>
     /virtual-devices@100/channel-devices@200/disk@1
```

2. 检索要用于ZFS池的设备的VDC实例：

```
[root@LDOM1 ~]#  cat /etc/path_to_inst
#
# Caution! This file contains critical kernel state
#
"/fcoe" 0 "fcoe"
"/iscsi" 0 "iscsi"
"/pseudo" 0 "pseudo"
"/scsi_vhci" 0 "scsi_vhci"
"/options" 0 "options"
"/virtual-devices@100" 0 "vnex"
"/virtual-devices@100/channel-devices@200" 0 "cnex"
"/virtual-devices@100/channel-devices@200/disk@0" 0 "vdc"
"/virtual-devices@100/channel-devices@200/pciv-communication@0" 0 "vpci"
"/virtual-devices@100/channel-devices@200/network@0" 0 "vnet"
"/virtual-devices@100/channel-devices@200/network@1" 1 "vnet"
"/virtual-devices@100/channel-devices@200/network@2" 2 "vnet"
"/virtual-devices@100/channel-devices@200/network@3" 3 "vnet"
"/virtual-devices@100/channel-devices@200/disk@1" 1 "vdc" << We want
this one
```

3. 编辑 `/platform/sun4v/kernel/drv/vdc.conf`：

```
block-size-list="1:4096";
```

这意味着为设备实例1分配的块大小为4096。

作为另一个示例、假设需要为vdsk实例1到6配置4K块大小和 `/etc/path_to_inst` 内容如下：

```
"/virtual-devices@100/channel-devices@200/disk@1" 1 "vdc"
"/virtual-devices@100/channel-devices@200/disk@2" 2 "vdc"
"/virtual-devices@100/channel-devices@200/disk@3" 3 "vdc"
"/virtual-devices@100/channel-devices@200/disk@4" 4 "vdc"
"/virtual-devices@100/channel-devices@200/disk@5" 5 "vdc"
"/virtual-devices@100/channel-devices@200/disk@6" 6 "vdc"
```

4. 最终版本 `vdc.conf` 文件应包含以下内容：

```
block-size-list="1:8192","2:8192","3:8192","4:8192","5:8192","6:8192";
```

小心
配置vdc.conf并创建vdsk后、必须重新启动LLOM。这一步是不可避免的。块大小更改仅在重新启动后生效。继续进行zpool配置、并确保将ashift正确设置为12、如上所述。

ZFS意图日志**(ZIL)**

通常，没有理由在其他设备上查找ZFS意图日志(ZIL)。日志可以与主池共享空间。单独的ZIL主要用于使用在现代存储阵列中缺少写入缓存功能的物理驱动器。

对数偏差

设置 `logbias` 用于托管Oracle数据的ZFS文件系统上的参数。

```
zfs set logbias=throughput <filesystem>
```

使用此参数可降低整体写入级别。在默认设置下、写入的数据会先提交到ZIL、然后再提交到主存储池。此方法适用于使用普通驱动器配置的配置、该配置包括基于SSD的ZIL设备和用于主存储池的旋转介质。这是因为它允许在可用延迟最低的介质上的单个I/O事务中进行提交。

如果使用的是具有自身缓存功能的现代存储阵列、则通常不需要使用此方法。在极少数情况下、可能需要将具有单个事务的写入提交到日志中、例如由高度集中且对延迟敏感的随机写入组成的工作负载。写入放大会产生一定的后果、因为记录的数据最终会写入主存储池、从而导致写入活动增加一倍。

直接**I/O**

许多应用程序(包括Oracle产品)都可以通过启用直接I/O来绕过主机缓冲区缓存此策略无法按预期用于ZFS文件系

统。尽管会绕过主机缓冲区缓存，但ZFS本身仍会继续缓存数据。在使用FIO或SIO等工具执行性能测试时、此操作可能会导致误导性的结果、因为很难预测I/O是到达存储系统还是在操作系统中本地缓存。此操作还会使使用此类综合测试来比较ZFS与其他文件系统的性能变得非常困难。实际上、在实际用户工作负载下、文件系统性能几乎没有差别。

多个**zpool**

必须在zpool级别对基于ZFS的数据执行基于Snapshot的备份、还原、克隆和归档、并且通常需要多个zpool。zpool类似于LVM磁盘组、应使用相同的规则进行配置。例如、数据库的布局可能最好是将数据文件驻留在上 `zpool1` 以及上的归档日志、控制文件和重做日志 `zpool2`。此方法允许使用标准热备份、其中数据库将置于热备份模式、然后是的快照 `zpool1`。然后、数据库将从热备份模式中删除、并强制执行日志归档和的快照 `zpool2` 已创建。还原操作需要卸载zfs文件系统并使zpool完全脱机、然后执行SnapRestore还原操作。然后、可以将zpool重新联机并恢复数据库。

filesystemio_options

Oracle参数 `filesystemio_options` 与ZFS的工作方式不同。条件 `setall` 或 `directio` 使用时、写入操作是同步的、并会绕过操作系统缓冲区缓存、但读取操作会由ZFS进行缓冲。此操作会导致性能分析出现困难、因为I/O有时会被ZFS缓存截获并提供服务、从而使存储延迟和总I/O比看起来要小。