附加说明
Enterprise applications

NetApp
May 09, 2024

目录

附加说明

Oracle数据库性能优化和基准测试过程

准确测试数据库存储性能是一个极其复杂的主题。它需要了解以下问题：

- IOPS和吞吐量
- 前台和后台I/O操作之间的区别
- 延迟对数据库的影响
- 许多操作系统和网络设置也会影响存储性能

此外、还需要考虑执行非存储数据库任务。有时、优化存储性能不会带来任何有用的优势、因为存储性能不再是性能的限制因素。

现在、大多数数据库客户都选择全闪存阵列、这就需要考虑一些额外的注意事项。例如、考虑在双节点AFF A900系统上进行性能测试：

- 如果读/写比率为80/20、则两个A900节点可以在延迟甚至超过150µs微秒之前提供超过100万次的随机数据库IOPS。这远远超出了大多数数据库当前的性能需求、因此很难预测预期的改进。存储将作为瓶颈在很大程度上被消除。
- 网络带宽日益成为性能限制的常见来源。例如、旋转磁盘解决方案通常会成为数据库性能的瓶颈、因为I/O延迟非常高。当全闪存阵列消除延迟限制后、障碍往往会转移到网络上。在虚拟化环境和刀片式系统中、这一点尤为明显、因为它们的真正网络连接很难直观地呈现出来。如果由于带宽限制而无法充分利用存储系统本身、则可能会使性能测试复杂化。
- 由于全闪存阵列的延迟显著缩短、因此通常无法将全闪存阵列与包含旋转磁盘的阵列进行性能比较。测试结果通常没有意义。
- 将峰值IOPS性能与纯闪存阵列进行比较通常不是一项有用的测试、因为数据库不受存储I/O的限制例如、假设一个阵列可以承受50万次随机IOPS、而另一个阵列可以承受30万次随机IOPS。如果数据库将99%的时间花在CPU处理上、则这种差异在实际环境中无关紧要。这些工作负载从不充分利用存储阵列的全部功能。相反、在整合平台中、峰值IOPS功能可能至关重要、在该平台中、存储阵列应加载到其峰值功能。
- 在任何存储测试中、始终考虑延迟和IOPS。市场上的许多存储阵列都声称IOPS达到了极高水平、但延迟会使这些IOPS在这种水平下毫无用处。全闪存阵列的典型目标为1毫秒标记。更好的测试方法不是测量可能的最大IOPS、而是确定在平均延迟超过1毫秒之前存储阵列可以承受的IOPS数。

Oracle自动工作负载存储库和基准测试

Oracle性能比较的黄金标准是Oracle自动工作负载存储库(Automatic Workload Repository、AWR)报告。

AWR报告有多种类型。从存储角度来看、是指通过运行生成的报告 `awrrpt.sql` 命令功能最全面、最有价值、因为它针对特定数据库实例、并包含一些详细的直方图、这些直方图可按延迟细分存储I/O事件。

比较两个性能阵列时、理想情况下需要在每个阵列上运行相同的工作负载、并生成一个准确针对该工作负载的AWR报告。如果工作负载运行时间非常长、则可以使用一个AWR报告、其中经过的时间包含开始和停止时间、但最好将AWR数据细分为多个报告。例如、如果批处理作业从午夜运行到早上6点、请创建一系列从午夜到凌晨1点、从凌晨1点到凌晨2点的一小时AWR报告、依此类推。

在其他情况下、应优化非常短的查询。最佳选择是基于查询开始时创建的AWR快照和查询结束时创建的第二

个AWR快照创建AWR报告。否则、数据库服务器应保持安静、以最大限度地减少后台活动、因为后台活动会掩盖正在分析的查询的活动。

> 如果AWR报告不可用、则Oracle statspack报告是一个很好的替代方案。它们包含与AWR报告大部分相同的I/O统计信息。

Oracle AWR和故障排除

AWR报告也是分析性能问题的最重要工具。

与基准测试一样、性能故障排除要求您精确测量特定工作负载。如果可能、请在向NetApp支持中心报告性能问题或与NetApp或合作伙伴客户团队合作购买新的解决方案时提供AWR数据。

提供AWR数据时、请考虑以下要求：

- 运行 `awrrpt.sql` 命令以生成报告。输出可以是文本或HTML。
- 如果使用Oracle Real Application Clusters (RAC)、请为集群中的每个实例生成AWR报告。
- 确定问题存在的具体时间。AWR报告的最长可接受用时通常为一小时。如果问题持续数小时或涉及多小时操作(例如批处理作业)、请提供多个涵盖要分析的整个期间的一小时AWR报告。
- 如果可能、将AWR快照间隔调整为15分钟。此设置允许执行更详细的分析。这还需要执行更多的 `awrrpt.sql` 以提供每15分钟间隔的报告。
- 如果问题是运行时间非常短的查询、请根据操作开始时创建的AWR快照和操作结束时创建的第二个AWR快照提供AWR报告。否则、数据库服务器应保持安静、以最大限度地减少后台活动、因为后台活动会掩盖所分析操作的活动。
- 如果在特定时间报告了性能问题、但在其他时间未报告、请提供其他证明性能良好的AWR数据以供比较。

CALIBRAT_IO

。 `calibrate_io` 切勿使用命令测试、比较存储系统或对其进行基准测试。如Oracle文档中所述、此操作步骤会校准存储的I/O功能。

校准与基准测试不同。此命令的目的是通过问题描述I/O来帮助校准数据库操作、并通过优化向主机发出的I/O级别来提高其效率。因为执行的I/O类型 `calibrate_io` 操作不代表实际的数据库用户I/O、结果不可预测、而且经常甚至无法重现。

SLOB2

SOLB2 (Song Little Oracle基准)已成为评估数据库性能的首选工具。它由Kevin Clsson开发、可从获取 "https://kevinclosson.net/slob/"。安装和配置只需几分钟、它会使用实际的Oracle数据库在用户可定义的表空间上生成I/O模式。它是少数几个可以使全闪存阵列的I/O饱和的测试选项之一此外、它还有助于生成低得多的I/O级别、以模拟IOPS低但对延迟敏感的存储工作负载。

Swingbench

Swingbench可用于测试数据库性能、但要以对存储造成压力的方式使用Swingbench、则极为困难。NetApp尚未从Swingbench中检测到任何测试产生足够的I/O来为任何AFF阵列带来大量负载。在有限情况下、可以使用订单输入测试(Order Entry Test、OOT)从延迟角度评估存储。如果数据库对特定查询具有已知的延迟依赖关系、则此功能可能会很有用。必须注意确保主机和网络配置正确、以实现全闪存阵列的潜在延迟。

HammerDB

HAMmerDB是一款数据库测试工具、用于模拟TPC-C和TPC-H基准测试等。构建一个足够大的数据集可能需要花费大量时间才能正确执行测试、但它可以作为一个有效的工具来评估OLTP和数据仓库应用程序的性能。

猎户座

Oracle ORION工具通常与Oracle 9一起使用、但尚未对其进行维护、以确保与各种主机操作系统中的更改兼容。由于与操作系统和存储配置不兼容、因此很少与Oracle 10或Oracle 11结合使用。

Oracle重新编写了该工具、默认情况下会随Oracle 12c一起安装。虽然此产品已得到改进、并使用了与实际Oracle数据库相同的许多调用、但它使用的代码路径或I/O行为与Oracle不同。例如、大多数Oracle I/O都是同步执行的、这意味着数据库会暂停、直到I/O完成、因为I/O操作在前台完成。简单地将随机I/O充斥存储系统并不是真正的Oracle I/O、也不提供比较存储阵列或衡量配置更改影响的直接方法。

尽管如此、也有一些适用于ORION的用例、例如、对特定主机-网络-存储配置的最大可能性能进行常规测量、或者对存储系统的运行状况进行评估。通过仔细测试、可以设计出可用的ORION测试来比较存储阵列或评估配置更改的影响、前提是这些参数包括考虑IOPS、吞吐量和延迟、并尝试忠实地复制真实的工作负载。

过时的NFSv3锁定和Oracle数据库

如果Oracle数据库服务器崩溃、则在重新启动时、陈旧的NFS锁定可能会出现问题。通过仔细注意服务器上的名称解析配置、可以避免此问题。

出现此问题的原因是、创建锁定和清除锁定使用两种略有不同的名称解析方法。其中涉及两个进程、即网络锁定管理器(Network Lock Manager、NLM)和NFS客户端。NLM使用 `uname n` 要确定主机名、请使用 `rpc.statd` 流程使用 `gethostbyname()`。这些主机名必须匹配、操作系统才能正确清除陈旧锁定。例如、主机可能正在查找属于的锁定 `dbserver5`，但主机已将锁定注册为 `dbserver5.mydomain.org`。条件 `gethostbyname()` 返回的值与不相同 `uname -a`，则锁定释放过程未成功。

以下示例脚本将验证名称解析是否完全一致：

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

条件 `gethostbyname` 不匹配 `uname`，可能是陈旧的锁定。例如、此结果揭示了一个潜在问题：

```
uname -n yields: dbserver5
gethostbyname yields: dbserver5.mydomain.org
```

通常、可以通过更改主机在中的显示顺序来查找解决方案 `/etc/hosts`。例如、假设主机文件包含以下条目：

```
10.156.110.201  dbserver5.mydomain.org dbserver5 loghost
```

要解析此问题描述、请更改完全限定域名和短主机名的显示顺序：

```
10.156.110.201  dbserver5 dbserver5.mydomain.org loghost
```

`gethostbyname()` 现在返回短 `dbserver5` 主机名、与的输出匹配 `uname`。因此、锁定会在服务器崩溃后自动清除。

Oracle数据库的WAFL对齐验证

正确对齐WAFL对于获得良好性能至关重要。尽管ONTAP以4 KB单位管理块、但这并不意味着ONTAP以4 KB单位执行所有操作。事实上、ONTAP支持不同大小的块操作、但底层记帐由WAFL以4 KB单位进行管理。

术语"对齐"是指Oracle I/O与这些4 KB单位的对应关系。要获得最佳性能、需要将一个Oracle 8 KB块驻留在驱动器上的两个4 KB WAFL物理块上。如果块偏移2 KB、则此块位于一个4 KB块的一半、一个单独的完整4 KB块、然后是第三个4 KB块的一半。这种排列会导致性能下降。

对齐不是NAS文件系统的问题。Oracle数据文件会根据Oracle块的大小与文件开头对齐。因此、8 KB、16 KB和32 KB的块大小始终对齐。所有块操作都会与文件开头偏移、以4 KB为单位。

与此相反、LUN通常在开始时包含某种类型的驱动程序标头或文件系统元数据、以创建偏移。在现代操作系统中、对齐很少会成为问题、因为这些操作系统专为可能使用本机4 KB扇区的物理驱动器而设计、这也需要将I/O与4 KB边界对齐以获得最佳性能。

但也有一些例外情况。数据库可能是从未针对4 KB I/O进行优化的旧版操作系统迁移的、或者分区创建期间的用户错误可能导致偏移量大小不以4 KB为单位。

以下示例是Linux专用的、但操作步骤可适用于任何操作系统。

已对齐

以下示例显示了对具有单个分区的单个LUN的对齐检查。

首先、创建使用驱动器上所有可用分区的分区。

```
[root@host0 iscsi]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF
disklabel
Building a new DOS disklabel with disk identifier 0xb97f94c1.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.
The device presents a logical sector size that is smaller than
the physical sector size. Aligning to a physical sector (or optimal
I/O) size boundary is recommended, or performance may be impacted.
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-10240, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-10240, default 10240):
Using default value 10240
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
[root@host0 iscsi]#
```

可以使用以下命令以数学方式检查对齐情况：

```
[root@host0 iscsi]# fdisk -u -l /dev/sdb
Disk /dev/sdb: 10.7 GB, 10737418240 bytes
64 heads, 32 sectors/track, 10240 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 65536 bytes
Disk identifier: 0xb97f94c1
   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1              32    20971519    10485744   83  Linux
```

输出显示单位为512字节、分区起始单位为32。这是总共32 x 512 = 16、834字节、是4 KB WAFL块的整数倍。此分区已正确对齐。

要验证是否正确对齐、请完成以下步骤：

1. 确定LUN的通用唯一标识符(UUID)。

```
FAS8040SAP::> lun show -v /vol/jfs_luns/lun0
              Vserver Name: jfs
                  LUN UUID: ed95d953-1560-4f74-9006-85b352f58fcd
                    Mapped: mapped`                 `
```

2. 在ONTAP控制器上输入节点Shell。

```
FAS8040SAP::> node run -node FAS8040SAP-02
Type 'exit' or 'Ctrl-D' to return to the CLI
FAS8040SAP-02> set advanced
set not found.  Type '?' for a list of commands
FAS8040SAP-02> priv set advanced
Warning: These advanced commands are potentially dangerous; use
         them only when directed to do so by NetApp
         personnel.
```

3. 对第一步中确定的目标UUID启动统计收集。

```
FAS8040SAP-02*> stats start lun:ed95d953-1560-4f74-9006-85b352f58fcd
Stats identifier name is 'Ind0xffffff08b9536188'
FAS8040SAP-02*>
```

4. 执行一些I/O使用非常重要 `iflag` 用于确保I/O是同步的且不缓冲的参数。

> 使用此命令时请格外小心。反转 `if` 和 `of` 参数会销毁数据。

```
[root@host0 iscsi]# dd if=/dev/sdb1 of=/dev/null iflag=dsync count=1000
bs=4096
1000+0 records in
1000+0 records out
4096000 bytes (4.1 MB) copied, 0.0186706 s, 219 MB/s
```

5. 停止统计信息并查看对齐直方图。所有I/O都应位于中 `.0` 存储分段、表示I/O与4 KB块边界对齐。

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffff08b9536188
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-
4f74-9006-85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.0:186%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.1:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.2:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.3:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.4:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.5:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.6:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.7:0%
```

未对齐

以下示例显示了未对齐的I/O：

1. 创建不与4 KB边界对齐的分区。这不是现代操作系统上的默认行为。

```
[root@host0 iscsi]# fdisk -u /dev/sdb
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First sector (32-20971519, default 32): 33
Last sector, +sectors or +size{K,M,G} (33-20971519, default 20971519):
Using default value 20971519
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

2. 创建分区时使用的是33扇区偏移、而不是默认的32扇区偏移。重复中所述的操作步骤 "已对齐"。直方图显示如下：

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffff0468242e78
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-
4f74-9006-85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.0:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.1:136%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.2:4%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.3:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.4:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.5:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.6:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.7:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_partial_blocks:31%
```

未对齐情况很明显。I/O大部分落在*中 *.1 存储分段、与预期偏移匹配。创建分区时、该分区会比优化默认值更远地移动到设备中512字节、这意味着直方图偏移512字节。

此外、还可以使用 read_partial_blocks 统计信息不为零、这意味着执行的I/O未填满整个4 KB块。

重做日志记录

此处介绍的过程适用于数据文件。Oracle重做日志和归档日志具有不同的I/O模式。例如、重做日志记录是对单个文件的循环覆盖。如果使用默认的512字节块大小、则写入统计信息如下所示：

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffff0468242e78
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-4f74-
9006-85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.0:12%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.1:8%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.2:4%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.3:10%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.4:13%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.5:6%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.6:8%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.7:10%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_partial_blocks:85%
```

I/O将分布在所有直方图分段中、但这不是性能问题。但是、使用4 KB块大小可能会有利于极高的重做日志记录速率。在这种情况下、需要确保重做日志记录LUN正确对齐。但是、这对于获得良好性能并不像数据文件对齐那样重要。