第 3 章

Exadata Smart FlashLog

传统架构的 Oracle 数据库的在线日志文件写性能直接影响数据库的整体性能，对 OLTP 类型的业务系统而言，这一点尤为明显。尽管用户会尽最大的努力以确保重做日志写入的 IO 优先级，但重做日志写入有时仍然会很慢，严重时可能会长达几秒！

当出现了高 IO 读写能力的存储设备（如 SSD 固态硬盘、PCI-E 闪存卡）后，通常建议将在线日志文件这一类文件优先放入这些高性能的存储设备中，以提升整个数据库性能。

3.1 FlashLog 工作原理

在第 2 个版本的 Exadata 中，Oracle 公司为了进一步拓宽 Exadata 一体机的客户群体，宣称 Exadata 除了适用于 OLAP 系统之外，也适用于 OLTP 系统。那怎么才能满足 OLTP 系统对存储系统的高 IOPS 要求呢？于是，Exadata 引入了 PCI-E 闪存卡。但当时 PCI-E 闪存卡的价格非常昂贵，同时存储空间非常有限，如何充分利用这些 PCI-E 闪存卡就成了一门艺术。最开始是将这些 PCI-E 闪存卡全部创建成 FlashCache，热点数据会自动缓存到 FlashCache 中，以此来解决热点数据的高 IOPS 需求。解决完热点数据的高 IOPS 需求后，新的性能问题又显现出来，那就是日志文件写性能开始成为 OLTP 架构的瓶颈。

于是，从 11.2.2.4 版本开始，Exadata 引入了 Exadata Smart Flash Logging 新特性（后文中简称为 FlashLog。注意，该特性要求数据库版本必须在 11.2.0.2BP11 及以上）。该特性的主要优化思想是从每个存储节点的 PCI-E 闪存卡中开辟出一小块区域，作为在线日志文件的辅助存放目的地。当接收到写入重做日志条目的请求时，Exadata 允许数据库的 LGWR 进程将重做日志条目同时并行写入 FlashLog 和机械磁盘上的在线日志文件。只要 FlashLog 和机械磁盘中的在线日志文件有任意一个最先完成重做日志写操作，就会通知 Oracle 数据库继续工作。该特性的引入明显改善了 Exadata 中重做日志写入的响应时间和吞吐量，日志文件写不再成为 Exadata 的性能瓶颈。

注意：FlashLog 并不是永久的重做日志存放目的地，它仅仅是一个临时的重做日志存放地，目的是加快重做日志写的响应时间，以改善整个 Oracle 数据库的性能。FlashLog 会一直存放着这些重做日志，直到这些重做日志被安全地写回到机械磁盘上的在线日志文件中。

FlashLog 会处理所有的实例崩溃和恢复场景，而不需要数据库管理员进行任何额外或特殊的干预。像传统的 Oracle 数据库从在线日志文件进行实例恢复一样，对最终的用户而言，临时存放在 FlashLog 中的重做日志如何帮助实例恢复，是一个完全透明的过程。

FlashLog 特性的工作原理如图 3.1 所示，可概括成以下步骤。

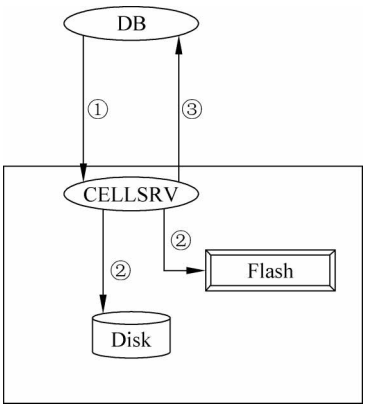

图 3.1 FlashLog 特性工作原理

（1）首先，Oracle 数据库在完成 commit 操作之前，需要将计算节点 Oracle 数据库 SGA 中 Log Buffer 中的重做日志写入在线日志文件中，所以计算节点向 Exadata 的存储节点发起重做日志写入请求。

（2）接着，Exadata 存储节点的 CELLSRV 进程同时向 FlashLog 和机械磁盘上的在线日志文件发出写入重做日志条目的操作。

（3）最后，机械磁盘上的在线日志文件写入操作和 FlashLog 上的写入操作，无论哪个最先完成，存储节点的 CELLSRV 进程都会认为重做日志写入操作已经完成。此刻就会通知计算节点，返回一个写入完成的响应，计算节点可以继续工作。

从 FlashLog 的工作原理可以看出，FlashLog 特性可以防止偶尔缓慢的机械硬盘 IO 响应，或者偶尔缓慢的 PCI-E 闪存卡 IO 响应给 Oracle 数据库带来性能问题。如图 3.2 所示为 FlashLog 特性关闭和开启时，重做日志写性能的 IO 响应对比。

可以看出，开启 FlashLog 特性后，基本上会完全消除重做日志写操作时出现异常的 IO 延时峰值，数据库运行会非常平稳，不会出现卡顿的现象。

FlashLog 是从每个存储节点的 PCI-E 闪存卡中开辟出的存储区域，它具有如下特点。

- FlashLog 存储区域在存储节点的所有 PCI-E 闪存卡中均匀分配，默认总大小为 512MB。
- FlashLog 存储区域像一个巨大的循环缓冲区，在将 FlashLog 存储区域相应的重做日志写入到机械磁盘上的在线日志文件后，才可以重新使用这些 FlashLog 存储区域。
- FlashLog 存储区域不是在线日志文件的镜像。换句话说，它并不是在线日志文件的完全备份。当磁盘的 IO 响应不及时，它只是临时地存放最新的重做日志条目。这种方式需要的 FlashLog 空间会比较小，可以剩余更多的 PCI-E 闪存空间留给 FlashCache 使用。

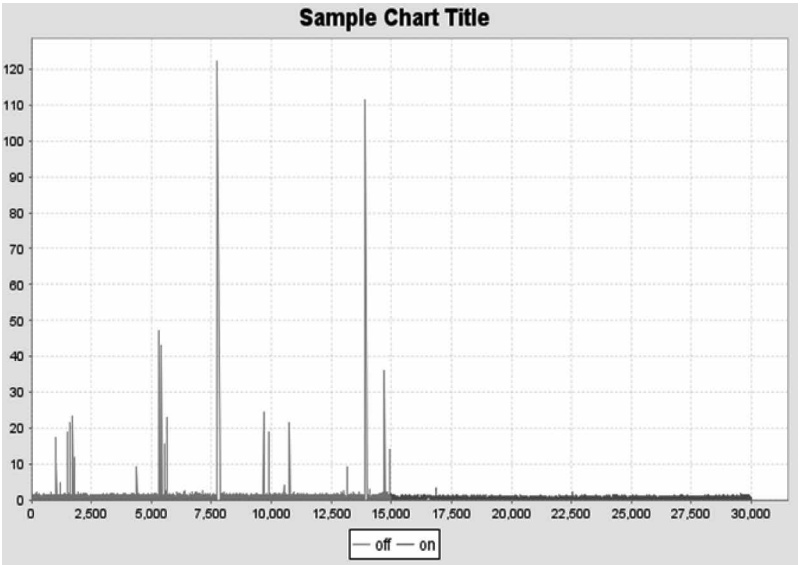

图 3.2 FlashLog 特性关闭与开启的性能对比

3.2 管理 FlashLog

在大多数情况下,每个存储节点默认分配的 512MB FlashLog 存储区域就已经足够了,但对极少数的系统而言,这是远远不够的。例如,数据库的重做日志产生率非常高,或者许多数据库实例整合到同一台 Exadata 上等,此时 512MB 的 FlashLog 存储区域可能就不足以支撑极速的重做日志写操作。在这种情况下,就需要手动增加 FlashLog 的大小。

3.2.1 FlashLog 日常管理

删除存储节点的 FlashLog,具体命令如下。

```
[root@sddxcel01 ~]# cellcli -e drop flashlog all
Flash log sddxcel01_FLASHLOG successfully dropped
[root@sddxcel01 ~]#
[root@sddxdb01 onecommand]# dcli -g ./cell_group -l root cellcli -e list flashlog
attributes name,size,status
sddxcel02: sddxcel02_FLASHLOG    512M    normal
sddxcel03: sddxcel03_FLASHLOG    512M    normal
[root@sddxdb01 onecommand]#
```

如果打算创建 FlashCache 和 FlashLog,则必须先创建 FlashLog,然后再创建 FlashCache。如果先创建 FlashCache,则会将所有的 PCI-E 闪存空间全部用来创建 FlashCache,导致没有多余的空间来创建 FlashLog。具体命令如下。

```
[root@sddxcel01 ~]# cellcli -e create flashlog all
Flash log sddxcel01_FLASHLOG successfully created
[root@sddxcel01 ~]#
[root@sddxcel01 ~]# cellcli -e create flashcache all
Flash cache sddxcel01_FLASHCACHE successfully created
```

查看 FlashLog 的详细信息,具体命令如下。

```
[root@dm02celadm01 ~]# cellcli -e list flashlog detail
         name:                   dm02celadm01_FLASHLOG
         cellDisk:
FD_00_dm02celadm01,FD_01_dm02celadm01,FD_02_dm02celadm01,FD_03_dm02celadm01
        creationTime:           2016-10-19T18:28:02+08:00
        degradedCelldisks:
        effectiveSize:          512M
        efficiency:             99.99785580691346
        id:                     ea422489-a82e-4fa7-877a-16b1c57ee63c
        size:                   512M
        status:                 normal
[root@dm02celadm01 ~]#
```

创建大小为 1GB 的 FlashLog,具体命令如下。

```
[root@dm02celadm01 ~]# cellcli -e CREATE FLASHLOG ALL SIZE=1G
```

3.2.2 关闭 FlashLog 特性

FlashLog 特性可以在存储节点层面关闭,也可以在数据库层面关闭。关闭 FlashLog 特性主要有如下 3 种方式。

(1) 在数据库层面关闭 FlashLog 特性。数据库从 11.2.0.2BP11 版本之后,就默认自动地开启了 FlashLog 特性,可能通过修改隐含参数的方式来手动关闭 FlashLog 特性。

如果数据库版本为 11.2.0.2,则通过以下命令来关闭 FlashLog 特性。

```
SQL> alter system set "_third_spare_parameter" = 0 scope=spfile sid='*';
```

如果数据库版本为 11.2.0.3 或之上,则通过以下命令来关闭 FlashLog 特性。

```
SQL> alter system set "_enable_flash_logging" = false scope=spfile sid='*';
```

(2) 在存储节点层面直接删除 FlashLog。这种方式将导致整个 Exadata 上的所有数据库都无法使用 FlashLog 特性。

(3) 除此之外,还可以使用IORM来控制FlashLog特性的开启或关闭。Exadata的IO资源器(IORM)有所增强,足以控制不同数据库的FlashLog特性是否关闭或开启,将FlashLog的宝贵资源留给那些非常重要的核心生产库使用,见如下代码。

```
CellCLI > ALTER IORMPLAN                                              -
          dbPlan = (                                                  -
                (name = prod, flashcache = on, flashLog = on),        -
                (name = dw, flashcache = on, flashLog = on),          -
                (name = prod_test, flashcache = off, flashLog = off), -
                (name = prod_dev, flashcache = off, flashLog = off)   -
                (name = other, flashcache = on, flashLog = on))
```

注意:对绝大部分的系统而言,强烈建议开启FlashLog特性,只有那种只读的数据库或测试数据库,才可以尝试关闭FlashLog特性。

3.3 FlashLog诊断

为便于日志文件写相关的性能分析,有时需要将FlashLog的内容转储成文件。在对存储节点的CELLSRV进程收集statedump信息时,大量的FlashLog信息也会写入跟踪文件中。主要包括以下信息。

- 整个FlashLog的统计信息。
- 每个单独的FlashLog存储区域的统计信息。
- FlashLog队列的工作列表。
- 详细的日志文件写的IO直方图信息,包括写磁盘的IO直方图和写闪存的IO直方图。

除了对存储节点的CELLSRV进程收集statedump信息会同时收集FlashLog信息之外,还可以在cellcli命令行中设置FlashLog事件。FlashLog事件具体的语法如下。

```
Cellcli > alter cell events = "immediate cellsrv.cellsrv_flashlog('action', 'str')"
```

其中action参数允许的几个参数值及相应的说明如表3.1所示。

表3.1 action参数值说明

action参数值	action参数值说明
clear_metrics	会将所有的FlashLog相关的指标全部置空。该方式仅适用于基准测试的情景
delete_saved_redo	会将指定的Celldisk上保存的重做日志全部删除。这种做法非常危险,仅仅用于测试目的

续表

action 参数值	action 参数值说明
dump_basic	打印输出 FlashLog 的基本信息
dump_store	指定一个 Flashdisk，会将该 Flashdisk 上 FlashLog 的内容给 dump 出来，但不会将重做日志自身的内容给 dump 出来，只是将每一个重做日志条目的 RBA 号和重做日志的 sequence # 号给 dump 出来

例如，将存储节点中 FlashLog 的信息转储成 trace 文件，具体命令如下。

```
CellCLI > alter cell events = "immediate cellsrv.cellsrv_flashlog('dump_basic',0)"
Dump sequence # 1 has been written to /opt/oracle/cell12.1.2.3.3_LINUX.X64_161013/log/
diag/asm/cell/cell03/trace/svtrc_6453_57.trc
Cell cell03 successfully altered

CellCLI >
```

此时，就会将该存储节点中 FlashLog 的相关信息转储到 svtrc_6453_57.trc 文件中。

3.4 监控 FlashLog 性能

当数据库出现性能问题，尤其是重做日志写方面的性能问题时，一定要关注 FlashLog 特性是否正常工作。

3.4.1 FlashLog 性能指标

Exadata 存储软件有一项性能指标，它记录了所有对象的统计信息。将对象类型设置为 FLASHLOG 时，就会展示出 FlashLog 特性相关的指标。具体命令如下。

```
CellCLI > list metriccurrent where objecttype = 'FLASHLOG';
         FL_ACTUAL_OUTLIERS                FLASHLOG         0 IO requests
         FL_BY_KEEP                        FLASHLOG         0
         FL_DISK_FIRST                     FLASHLOG         0 IO requests
         FL_DISK_IO_ERRS                   FLASHLOG         0 IO requests
         FL_EFFICIENCY_PERCENTAGE          FLASHLOG         100 %
         FL_EFFICIENCY_PERCENTAGE_HOUR     FLASHLOG         100 %
         FL_FLASH_FIRST                    FLASHLOG         0 IO requests
         FL_FLASH_IO_ERRS                  FLASHLOG         0 IO requests
         FL_FLASH_ONLY_OUTLIERS            FLASHLOG         0 IO requests
         FL_IO_DB_BY_W                     FLASHLOG         0.000 MB
         FL_IO_DB_BY_W_SEC                 FLASHLOG         0.000 MB/sec
         FL_IO_FL_BY_W                     FLASHLOG         0.000 MB
         FL_IO_FL_BY_W_SEC                 FLASHLOG         0.000 MB/sec
```

```
         FL_IO_W                    FLASHLOG        0 IO requests
         FL_IO_W_SKIP_BUSY          FLASHLOG        0 IO requests
         FL_IO_W_SKIP_BUSY_MIN      FLASHLOG        0.000 IO/sec
         FL_IO_W_SKIP_LARGE         FLASHLOG        0 IO requests
         FL_IO_W_SKIP_NO_BUFFER     FLASHLOG        0 IO requests
         FL_PREVENTED_OUTLIERS      FLASHLOG        0 IO requests
CellCLI>
```

FlashLog 特性相关的指标说明在后续章节中有详细介绍，这里不再进行解释。

3.4.2 FlashLog 性能分析

FlashLog 特性仅仅是影响到 log file parallel write 等待事件的响应时间，而不是影响到整个 log file sync 等待事件的响应时间。用户可以检查 AWR 报告中 log file parallel write 等待事件的统计信息，理想状态是 log file parallel write 等待事件的响应时间很少出现大于 32ms 的情况，并且不会出现大于 0.5s 的情况。可以从 AWR 报告的 Wait Event Histogram 部分获取到 log file parallel write 等待事件的统计信息，如图 3.3 所示。

		% of Waits							
Event	Total Waits	<1ms	<2ms	<4ms	<8ms	<16ms	<32ms	<=1s	>1s
log file parallel write	3076.1K	99.9	.0	.0	.0	.0			

图 3.3 log file parallel write 等待事件的统计信息

如果 log file parallel write 等待事件的响应时间比较长，则需要检查以下关于 FlashLog 的重要性能指标。

- FL_ACTUAL_OUTLIERS
- FL_DISK_IO_ERRS
- FL_FLASH_IO_ERRS
- FL_IO_W_SKIP_BUSY
- FL_IO_W_SKIP_LARGE
- FL_IO_W_SKIP_NO_BUFFER

正常情况下，FL_ACTUAL_OUTLIERS 指标值应该为 0，0 表示没有出现日志文件写延时超过 0.5s 的情况。如果该值≥1%的 FL_IO_W 指标值，则表明机械磁盘和 PCI-E 闪存卡的 IO 都非常糟糕。也许是某方面的配置不合理，也许是硬件存在性能问题，也有可能是系统非常繁忙。如果 FL_DISK_IO_ERRS 指标值不为 0，则表明在线日志文件所在的机械磁盘存在质量问题。如果 FL_FLASH_IO_ERRS 指标值不为 0，则表明 PCI-E 闪存卡存在质量问题。如果 PCI-E 闪存卡太繁忙，则需要检查所有存储节点中 PCI-E 闪存的 IO 负载情况。如果存储节点出现了 PCI-E 闪存 IO 过载的现象，则需要检查是什么原因导致了 PCI-E 闪存 IO 过载。

如果 FL_IO_W_SKIP_BUSY 指标值不为 0，则表明在线日志文件所在的磁盘太慢，或

产生的重做日志太多，以至于 FlashLog 中还有大量的重做日志未及时写回在线日志文件中（FlashLog 太繁忙，没办法及时响应新的重做日志写入操作）。通过查看 FL_ACTUAL_OUTLIERS 指标值，可以判断是否在线日志文件所在的磁盘太慢；通过查看 FL_IO_DB_W_SEC 和 FL_IO_FL_W_SEC 指标值，可以判断是否产生的重做日志太多。如果是重做日志太多，磁盘的 IO 吞吐量无法满足重做日志的写入量，则 FlashLog 特性无法解决该问题，只能增加更多的存储节点，以增加磁盘的 IO 吞吐量；或者想办法减少重做日志的数量。如果是磁盘太慢，则需要检查所有存储节点的 IO 负载情况。如果存储节点出现了 IO 过载的现象，则需要检查是什么原因导致了 IO 过载。

如果 FL_IO_W_SKIP_LARGE 指标值不为 0，则表明存储节点上默认的 512MB FlashLog 空间太小，需要增大 FlashLog 的大小。同样，如果 FL_IO_W_SKIP_NO_BUFFER 指标值比较大，也表明 FlashLog 缺少可用的缓冲区，因此重做日志写入时绕过了 FlashLog 区域。此时需要增大 FlashLog 的大小。具体增加 FlashLog 区域大小的命令如下。

```
CELLCLI > DROP FLASHLOG
CELLCLI > DROP FLASHCACHE
CELLCLI > CREATE FLASHLOG all size = 1G(using a new larger size)
CELLCLI > CREATE FLASHCACHE all
```

从以上代码中可以看到，同时也重建了 FlashCache。增大了 FlashLog 区域空间，自然就需要缩小 FlashCache 空间，因为两者共享 PCI-E 闪存卡资源。

从 FlashLog 的性能指标中还可以看到 FL_DISK_FIRST 和 FL_FLASH_FIRST 这两个性能指标，但切记它们不能作为判断机械磁盘和 PCI-E 闪存性能好坏的依据。

```
[root@dm02dbadm01 ~]# dcli -g cell_group -l root 'cellcli -e list metriccurrent
FL_DISK_FIRST,FL_FLASH_FIRST'
dm01cel01: FL_DISK_FIRST       FLASHLOG       9,831,747 IO requests
dm01cel01: FL_FLASH_FIRST      FLASHLOG       113,803 IO requests
dm01cel02: FL_DISK_FIRST       FLASHLOG       8,790,483 IO requests
dm01cel02: FL_FLASH_FIRST      FLASHLOG       113,700 IO requests
dm01cel03: FL_DISK_FIRST       FLASHLOG       7,963,935 IO requests
dm01cel03: FL_FLASH_FIRST      FLASHLOG       84,688 IO requests
```

从以上信息可以看出，机械磁盘最先完成重做日志写，完成响应的次数远远高于 PCI-E 闪存的次数，但这并不意味着机械磁盘的性能比 PCI-E 闪存卡的性能好。之所以出现这样的指标数据，仅仅是因为机械磁盘控制器上也有 Cache 缓存。正常情况下，写入到机械磁盘控制器上的 Cache 缓存就表示数据已经写入完成。所以，磁盘最先完成重做日志写，完成响应的次数远远高于 PCI-E 闪存的次数。