

Exadata 上的 Writeback 和 Writethrough

第三篇 V2 和 X2 上的 Flash Card

(www.lunar2013.com)

从 V2 和 X2 的机器上硬件带有 4 块具有写缓存功能 Sun Flash Accelerator F20 PCIe Card 卡，每个 F20 上有 4 个 FDOM 模块，每个 FDOM 模块是 24G（每个 F20 的容量为 $24 \times 4 = 96\text{GB}$ ）。因此 V2 和 X2 的每台存储服务器上可以支持 $4 \times 96\text{GB} = 384\text{GB}$ 的缓存，如图：



Sun Flash Accelerator F20 PCIe Card



每个 F20 上有 4 个 FDOM (Flash Disk On Module)

这个是 Exadata 上的 Sun Flash Accelerator F20 PCIe Card，其实际上是采用了 LSI 的 SAS HBA，这款 Accelerator F20 PCIe 最早应用于 Oracle 在 2009 年发布的 Exadata V2 上，从而将一体机从 DW 领域扩展到 OLTP 领域。

Exadata 上从 Image 11.2.3.2.0 (X3 的初始版本, X2 以前的机器可以将软件升级到这个版本) 开始, 支持 Flash Cache 的 WriteBack, 而此前的软件版本是 Writethrough。具体的信息已经在以前的 blog 中讲过, 这里不再赘述:

[Exadata 的磁盘自动管理-1-教你读懂 cell alert 中的磁盘相关信息](#)

[Exadata 的磁盘自动管理-2-Cell 上各种磁盘相关的概念](#)

[Exadata 的磁盘自动管理-3-磁盘自动管理的操作规则](#)

我这里只想分析一下, 不论是 Writeback (“写回”功能) 还是 Writethrough (“透写”功能), 他们的实现具体的工作方式和硬件之间有啥关系。

我们都知道, 硬盘本身没有 WB 或者 WT 的功能, 但是在 Exadata 上的 Disk Controller BBU 可以帮助他实现这个功能, 这个在日志中有所体现:

```
dm02cel01: 16_1    2014-01-17T02:00:10+08:00          info          "The disk
controller battery is executing a learn cycle and may temporarily enter WriteThrough Caching mode
as part of the learn cycle. Disk write throughput might be temporarily lower during this time.
The flash drives are not affected. The battery learn cycle is a normal maintenance activity that
occurs quarterly and runs for approximately 1 to 12 hours. Note that many learn cycles do not
require entering WriteThrough caching mode. When the disk controller cache returns to the normal
WriteBack caching mode, an additional informational alert will be sent. Battery Serial Number :
6198 Battery Type : iBBU08 Battery Temperature : 41 C Full Charge
Capacity : 1304 mAh Relative Charge : 99 % Ambient Temperature : 23 C"
dm02cel01: 16_2    2014-01-17T07:31:50+08:00          clear         "All disk
drives are in WriteBack caching mode. Battery Serial Number : 6198 Battery
Type : iBBU08 Battery Temperature : 45 C Full Charge Capacity :
1285 mAh Relative Charge : 54 % Ambient Temperature : 23 C"
```

正常情况下, 在 Writeback 模式, 数据会写入 Flash Card 的缓存, 但是在电池充放电时, 会自动改为 Writethrough 模式, 这样是为了防止异常断电时丢失数据。

比如, 假设在 Writeback 模式下工作, 当机器突然掉电了, 那么只要电池有电, 数据就可以被存储在缓存中, 那么在机器加电时, 数据就会从缓存直接写到 Flash Card, 保证了数据没有丢失。

对于普通磁盘来说, 在 Exadata 上是 “Disk Controller BBU” 来完成上述的包含工作, 而传统存储则有自己的 cache 系统完成, 在各种 SSD 或者说 FlashCard 上, 比如 Exadata V2 引入的闪存卡的硬件, 则是通过如下部件完成的:

96GB PCI Express Flash Accelerator F20 SAS HBA

- 24GB Solid State Flash Memory Module
- Energy Storage Module (ESM)
- PCI Express Flash Board

96GB PCI Express Flash Accelerator F20 M2 1.1 SAS HBA

- Energy Storage Module (ESM)
- 24GB Solid State Flash Memory Module, FW D20Y
- PCI Express Flash Board

Energy Storage Module (ESM) ----- 类似电源的模块
24GB Solid State Flash Memory Module, FW D20Y ----- 内存模块

注意, Disk Controller BBU 需要定期的充放电和学习, 这个在以前的 blog 中已经写过(《[如何看待 exadata 的 cell 节点出现的 writethrough/wirteback 模式更换或者控制器充放电信息](#)》), 这里不再赘述, 上面的一段 alert 日志信息就是充放电的过程。

Exadata 上, V2 和 X2 使用的 Accelerator F20 PCIe Card 采用的是基于 Flash 存储介质的固态硬盘。在硬件的列表清单中, 我发现 F20 卡上存在 Energy Storage Module (ESM) 和 24GB Solid State

Flash Memory Module, FW D20Y 。有了这个硬件设备,就能配合 Oracle 数据库和 Exadata 存储服务器软件,确保在缓存只有经常访问的数据:

- (1) 自动跳过不会被频繁访问的数据,或避免对不适合放在高速缓存中的数据进行缓存(备份, mirrored copies 的数据, ASM rebalance 的数据, expdp 的导出数据等等)
- (2) 数据库自己可以感知哪些数据会被频繁访问,这就是我们一直说“SGA 二级缓存”的作用
- (3) 当然,用户可以根据自己需要进行微调,例如:

在 Exadata 初始的缺省安装情况下,我们都是这么考虑的:用 Flash Cache 作为上述的二级缓存,Onecommand 中相应的命令如下(这些都可以手工控制的):

```
cellcli -e create celldisk all
cellcli -e drop flashcache
cellcli -e create flashcache all
```

然后就可以使用“ALTER TABLE TABLENAME STORAGE (CELL_FLASH_CACHE KEEP)”等命令来缓存数据库对象了,可以将 table, index, partition 或者 LOB 列缓存在 Flash Cache 上。

这个命令的选项如下:

DEFAULT : 系统自动选择是否 cache
KEEP : 手工指定需要 cache 对象
NONE : 不 cache 该对象

KEEP 对象的优先级高于 DEFAULT 的对象,因此,当系统需要自动 cache 一个 DEFAULT 的对象数据时不会把任何 KEEP 对象的缓存数据挤出 flash cache。

同时,为防止 KEEP 对象太多的占用 flash cache,系统最多只允许 KEEP 对象的总容量不超过总 flash cache 的 80%。

另外,为了防止 KEEP 的对象没有被经常访问(浪费了宝贵的 flash cache)却无限期占用缓存,系统有一个老化策略可以定期的从 KEEP 中清理老化的对象。

具体的使用方法举例:

```
ALTER INDEX lunar STORAGE (CELL_FLASH_CACHE keep);
ALTER INDEX lunar STORAGE (CELL_FLASH_CACHE NONE);
```

```
ALTER table lunar STORAGE (CELL_FLASH_CACHE keep);
ALTER table lunar STORAGE (CELL_FLASH_CACHE NONE);
```

这里为什么先要先“drop flashcache”,然后在“create flashcache all”呢? 文档上是这样说的:

When an Exadata cell is installed, by default, the CREATE CELL command creates flash cell disks on all flash cards, and then creates Exadata Smart Flash Cache on the flash cell disks. To change the size of the Exadata Smart Flash Cache or create flash grid disks it is necessary to remove the flash cache, and then create the flash cache with a different size, or create the flash grid disks.

也就是说,缺省的安装时,使用 CREATE CELL 创建命令会在所有的 flash cards 上创建 flash cell disk,然后会在这些 flash cell disk 上创建 ESFC(Exadata Smart Flash Cache)。而我们再使用 Onecommand 的时候,一般把 Flash Card 直接用于做 Flash Cache,这样,就必须把“CREATE CELL ”带来的基于 flash cell disk 的方式改为直接使用 flash cache 的方式。

这么做有什么好处呢?

我们从文档上找到“Using flash disk technology would limit the performance since flash disks would be connected to a relatively slow disk controller significantly reducing

the IOPS and bandwidth that could be delivered. ”。也就是说这种用法绕过了速度相对较低的 disk controller 。

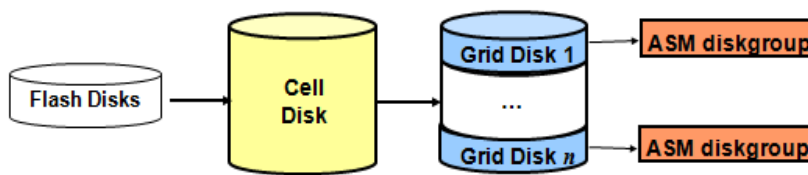
这里我们看到了 ESFC(Exadata Smart Flash Cache)的两种创建方式，即：

- 1, 先创建 flash disk, 然后在 flash disk 上创建 flash cache。
- 2, 直接使用 create flashcache all 命令创建 flash cache

现在我们来了解下 Flash disk 的其他用途：

Flash Cache

Flash-based grid-disks



在 Exadata 上, Flash Card 有两种用途: Smart Flash Cache (上面已经讲过) 和 Flash-based grid disks (比如作为 ASM DISKGROUP) 两种用途。

创建方式跟 cell 上普通的 HDD 的方式相差不多, 只是没有了每个 cell 前两块盘需要单独划分一部分空间用做系统盘的却别, Flash Disk 的划分跟 cell 的后面 10 块盘的使用方式一样。

Flash-based cell disks and grid disks

```

CellCLI> LIST CELLDISK DETAIL
name:          FD_00_cell101
diskType:      FlashDisk
. . .
name:          CD_00_cell101
diskType:      HardDisk
. . .

CellCLI> CREATE GRIDDISK ALL FLASHDISK -
PREFIX='FAST', SIZE=10G
GridDisk FAST_FD_00_cell101 successfully created
GridDisk FAST_FD_01_cell101 successfully created
. . .

CellCLI> LIST LUN WHERE cellDisk='FD_00_cell103' DETAIL
name:          1_1
cellDisk:      FD_00_cell103
deviceName:    /dev/sdn
diskType:      FlashDisk
physicalDrives: [9:0:2:0]

CellCLI> LIST PHYSICALDISK '[9:0:2:0]' DETAIL
name:          [9:0:2:0]
physicalFirmware: D20R
slotNumber:    "PCI Slot: 1; FDOM: 1"
  
```

一直以来, 很多玩 Exadata 的朋友都意识到一个问题, 为什么在 Exadata 上很少有人用 Flash card 做 ASM disk?

我猜这里有两个原因:

- 1, 在 Image 11.2.3.2.0 以前的版本, 对于 Flash disk 没有 WriteBack 的功能, 因此对于需要大量写入的场景, Flash disk 优势不大

2, 考虑到 ASM 中, 同一个磁盘组中的 Failure Group 的大小最好是一致的 (官方推荐), 那么可以想象, 此时的 F20 的容量太小, 难以发挥很大作用。

比如, 有些非 Exadata 的环境下, 我们通常会使用 Flash Card 建立一个 ASM Failour Group, 然后用 HDD 盘建立一个同样大小的 ASM Failour Group, 然后用他们做一个 ASM DG, 这样数据同时在 HDD 和 SSD (Flash Card) 上都有一份, 既享受了 Flash 带来的高速, 也享受了 HDD 永久化的安全保障, 同时还能满足 SSD 假死时的性能互补。

这个设计在 Exadata 的 V2 和 X2 上基本上很不现实了, 原因就是这两款机器的 F20 上那 4 个 FMOD, 每个 FMOD 只有 23G, 创建这样小的 ASM DISK 基本上意义不大。