鲲鹏 BoostKit 数据库使能套件

NVMe SSD 原子写 特性指南

文档版本 02

发布日期 2023-07-25

华为技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：
https://www.huawei.com/cn/psirt/vul-response-process
如企业客户须获取漏洞信息，请参见如下网址：
https://securitybulletin.huawei.com/enterprise/cn/security-advisory

目 录

1 特性介绍

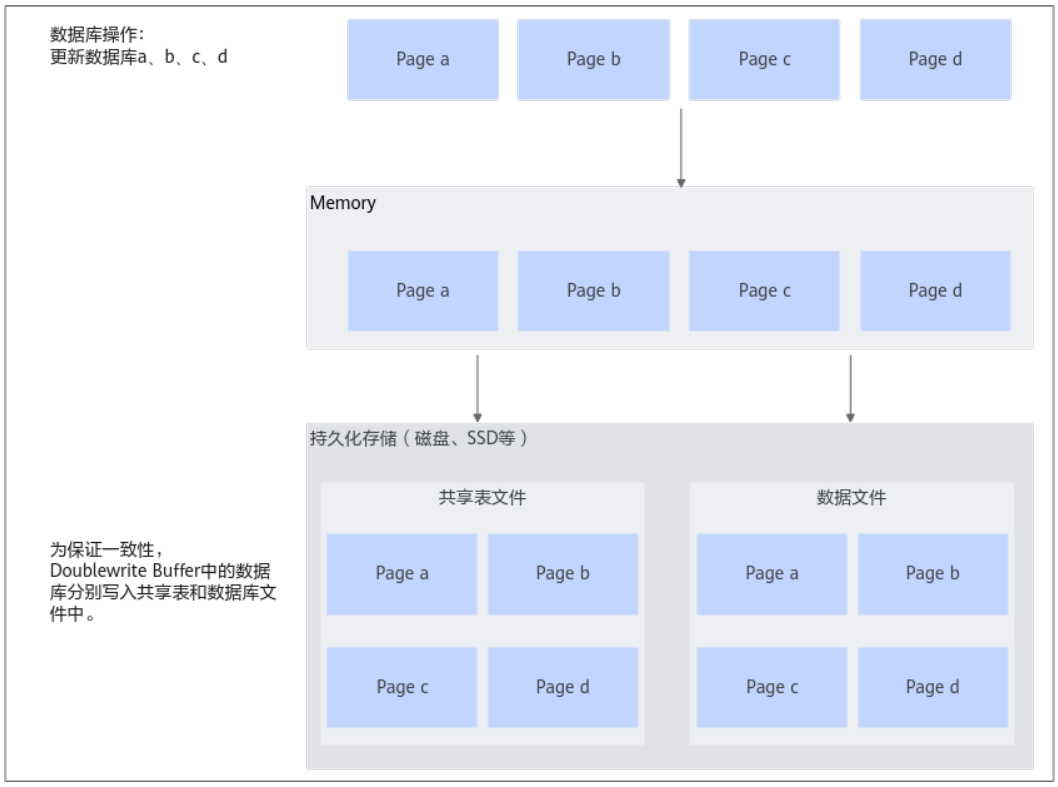
MySQL 数据库写操作介绍

DBMS（Database Management System）简称数据库，在当前互联网、金融等行业中获得了广泛的应用。

数据库事务（Database Transaction）是单个逻辑工作单元执行的一系列操作，要么完全地执行，要么完全地不执行。事务处理可以确保除非事务性单元内的所有操作都成功完成，否则不会永久更新面向数据的资源。

为保证上述事务的一致性和原子性，防止事务提交到持久化存储时出现不完整的情况，如因为突然断电、数据库或者操作系统挂死等意外情况导致部分数据库写入，部分未写入的情况，很多主流数据库采用了Doublewrite机制，如图1-1所示。

图 1-1 数据库写操作流程：两次写



由上述典型数据库写操作机制分析可知，数据库采用两次写的机制来保证事务的原子性，这会带来相同数据的两次写操作。

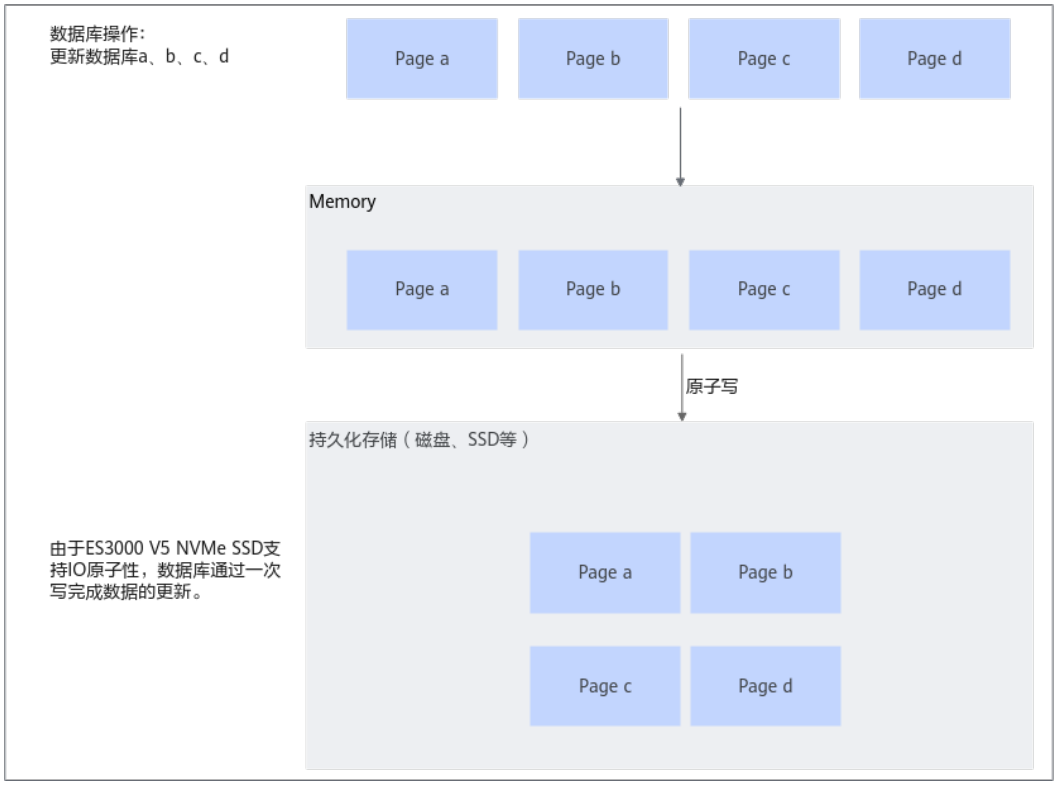
NVMe SSD 原子写特性实现原理

由前述可知，数据库为保证数据持久化到存储中不发生部分数据更新而导致的不一致问题，保障数据库的原子性，采用了Doublewrite机制，造成对存储的两次写操作。

针对这个问题，华为ES3000 V5 NVMe SSD提供原子写特性，保障写入ES3000 V5 NVMe SSD的IO操作的原子性，即一个IO要么完整的写入，要么整个写失败，不会出现一个IO中部分数据写入，部分未写入的情况，实现原理如图1-2所示。

这样数据库可不采用Doublewrite机制，也能保证数据完整落盘，减少一次数据写入操作，从而提升性能。

图 1-2 华为 ES3000 V5 原子写特性基本原理



本文将详细介绍数据库解决方案MySQL数据库场景下，使能华为新一代NVMe PCIe固态硬盘ES3000 V5的原子写特性的操作指导。

版本说明

本特性随Kunpeng Computing DC Solution 20.0.3版本发布。

2 环境要求

硬件要求

硬件要求如表2-1所示。

表 2-1 硬件要求

项目	说明
服务器	鲲鹏服务器
CPU	鲲鹏920处理器
硬盘要求	OS盘：900G SAS HDD/ RAID 1 (仅做推荐，至少需要两块盘) MySQL数据盘：华为ES3000 V5 NVMe SSD（固件版本最低要求 2151）

操作系统要求

操作系统要求如表2-2所示。

表 2-2 操作系统要求

项目	版本
CentOS	7.6 for Arm

说明

如果是全新安装操作系统，可选择“Minimal Install”安装方式并勾选Development Tools套件，否则很多软件包需要手动安装。

3 配置 NVMe SSD 原子写特性

在Linux发行版操作系统中，ext4文件系统支持MySQL 16KB page size的原子写入要求，通过BigAlloc选项可以更大粒度组织文件逻辑地址映射。

3.1 安装 NVMe SSD 驱动

步骤1 安装ES3000 V5 SSD驱动和NVMe卡管理工具。

- 安装ES3000 V5 SSD驱动的具体操作请参见**ES3000 V5 NVMe PCIe SSD 用户指南-安装驱动**。
- 安装hioadm NVMe卡管理工具的具体操作请参见**ES3000 V5 NVMe PCIe SSD 用户指南-安装工具包**。

说明

如果只使用ES3000 V5 NVMe PCIe SSD，不使用其他厂家的SSD，可以使用操作系统自带的NVMe驱动（不需要安装华为NVMe驱动）。

步骤2 驱动安装完成后，查询指定的SSD设备的固件版本，确认固件版本为ES3000 V5 2151及之后的固件版本。

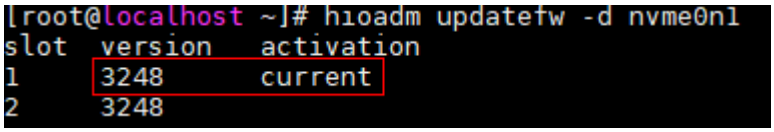
命令格式为：

hioadm updatefw -d <device>

其中device为待查询的SSD设备名称，例如“nvme0n1”。

使用实例：

```
hioadm updatefw -d nvme0n1
```



说明

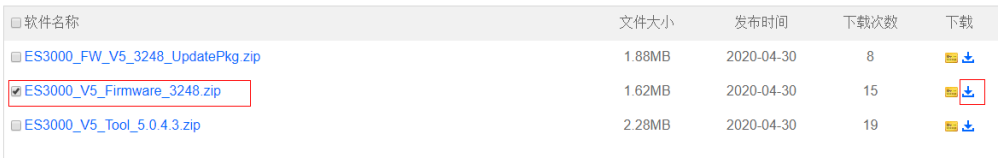
如果版本不符合要求，请参考**步骤3**升级固件版本。否则，跳过升级NVMe固件版本步骤。

步骤3 升级NVMe固件版本。

1. 打开**技术支持网站**，并搜索ES3000 V5。

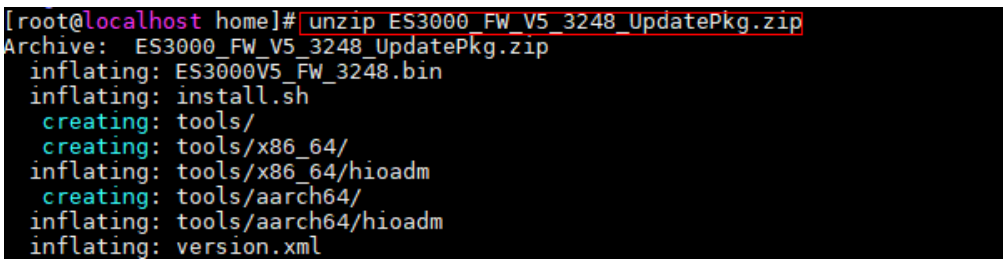
2. 点击“软件”标签，并且选择规划的固件版本包（这里推荐最新的固件版本包，且以最新的固件版本包为例截图说明）。

3. 选择固件升级包，并下载。



4. 上传到服务器“/home”目录，并解压。

unzip ES3000_FW_V5_3248_UpdatePkg.zip



5. 在“/home”目录下执行命令，升级NVMe固件版本。

命令格式为：

hioadm updatefw -d devicename -f fwimagefile [-s slot] [-a activeflag]

命令中参数详解如**表3-1**所示。

表 3-1 参数详解

参数	参数说明	取值
devicename	待升级的SSD设备名称	例如“nvme0”。
fwimagefile	目标固件镜像文件路径	例如“/home/fw_image.img”。
slot	目标固件镜像的槽位号	2、3 – slot 1是只读固件，不允许用户修改。 – 当未设置改选项时，默认升级非当前云行的slot。 – 如果当前运行的是slot 1，则缺省时默认升级slot 2。
activeflag	固件的激活方式	– 0：下载完固件后，设备会在下次复位时激活新的固件版本。 – 1：下载完固件后，设备会直接激活新的固件版本，无需等待下次复位。 **说明** 如果用户没有设置该选项，那么在下载完固件后，设备将在下次复位时自动激活新的固件版本。如果下载的固件版本与当前运行的版本相同，则激活状态将显示为“current”。

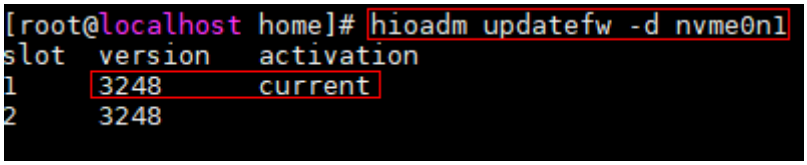
本文以如下命令为例，在“/home”目录下进行执行操作：

```
cd /home
hioadm updatefw -d nvme0n1 -f ES3000V5_FW_3248.bin -s 1 -a 1
```

输入**Y**，确定，回车。

6. 查看升级后的固件版本，确认已经升级成功。

```
hioadm updatefw -d nvme0n1
```



----结束

3.2 使能 NVMe SSD 原子写特性

步骤1 查询NVMe SSD原子写的使能状态。

命令格式为：

hioadm atomicwrite -d <device>

其中device为指定的SSD设备名称，例如“nvme0n1”。

使用实例：

```
hioadm atomicwrite -d nvme0n1
```

回显信息显示如下，表示原子状态为关闭状态。

```
atomic write status: Disabled.
```

步骤2 使能NVMe SSD原子写特性。

命令格式为：

hioadm atomicwrite -d <device> -f <value>

其中：

- device为指定的SSD设备名称，例如“nvme0n1”。
- value表示原子写开关使能。0代表关闭原子写；1代表开启原子写。

使用实例：

```
hioadm atomicwrite -d nvme0n1 -f 1
```

回显信息显示如下，表示开启原子写成功。

```
Enabling atomic write succeeded.
```

----结束

3.3 格式化 SSD

在Linux发行版操作系统中，ext4文件系统支持MySQL 16KB page size的原子写入要求，通过BigAlloc选项可以更大粒度组织文件逻辑地址映射。

您可以使用BigAlloc选项的mkfs.ext4命令来格式化NVMe SSD。

执行如下命令格式化NVMe SSD。命令格式为**mkfs.ext4 -O bigalloc -C 16384 <device>**，其中device为指定的SSD设备名称，例如“**/dev/nvme0n1**”。

```
mkfs.ext4 -O bigalloc -C 16384 /dev/nvme0n1
```

回显信息显示如下，表明已完成格式化NVMe SSD。


```
[root@localhost ~]# mkfs.ext4 -O bigalloc -C 16384 /dev/nvme0n1
mke2fs 1.42.9 (28-Dec-2013)

Warning: the bigalloc feature is still under development
See https://ext4.wiki.kernel.org/index.php/Bigalloc for more information

Discarding device blocks: done
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Cluster size=16384 (log=4)
Stride=0 blocks, Stripe width=0 blocks
195362816 inodes, 781404240 blocks
39070212 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
5962 block groups
131072 blocks per group, 32768 clusters per group
32768 inodes per group
Superblock backups stored on blocks:
        131072, 393216, 655360, 917504, 1179648, 3276800, 3538944, 6422528,
        10616832, 16384000, 31850496, 44957696, 81920000, 95551488, 286654464,
        314703872, 409600000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@localhost ~]#
```

4 安装与配置 MySQL 数据库

4.1 安装 MySQL 数据库

安装MySQL数据库的详细步骤，请参见《**MySQL 安装指南**》。

说明

格式化硬盘的操作步骤请按照本文档中的**3.3 格式化SSD**执行。

4.2 关闭 MySQL 数据库 Doublewrite 特性

修改[mysqld]参数后，需要重启数据库使参数生效。

步骤1 打开数据库配置文件。在本例中，配置文件路径为“/etc/my.cnf”。

```
vim /etc/my.cnf
```

步骤2 按“i”进入编辑模式，找到以下参数并进行修改。如果没有以下参数，则将该参数添加到文件中。

```
[mysqld]
innodb_flush_method=O_DIRECT
innodb_doublewrite=0
```

步骤3 按“Esc”键，输入**:wq!**，按“Enter”保存并退出编辑。

步骤4 重启数据库使参数生效。

```
service mysql restart
```

说明

启动数据库具体命令以《**MySQL 安装指南**》中不同安装方式下的相应启动方式为准。

步骤5 验证原子写特性配置是否成功。

1. 执行以下命令查询原子写状态。

```
hioadm atomicwrite -d nvme0n1
```

回显信息显示如下，表示原子写状态已开启。


```
[root@localhost ~]# hioadm atomicwrite -d nvme0n1
atomic write status: Enabled.
```

2. 数据库内确认“doublewrite”和“flush_method”参数是否已修改成功。

```
show variables like '%flush_method%';
show variables like '%doublewrite%';
```

```
mysql> show variables like '%flush_method%';
+---------------------+----------+
| Variable_name       | Value    |
+---------------------+----------+
| innodb_flush_method | O_DIRECT |
+---------------------+----------+
1 row in set (0.01 sec)

mysql> show variables like '%doublewrite%';
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| innodb_doublewrite | OFF   |
+--------------------+-------+
1 row in set (0.00 sec)
```

----结束

A 修订记录

发布日期	修订记录
2023-07-25	第二次正式发布。 优化各章节中的操作步骤表达。
2020-07-13	第一次正式发布。