数据库课程设计环境配置报告

18341035 王宇春

一、 模拟 NVM 环境

1. 检查内核是否支持 DAX 和 PMEM

```
root@LancerW:~# egrep '(DAX|PMEM)' /boot/config-5.8.0-33-generic -r
CONFIG_X86_PMEM_LEGACY_DEVICE=y
CONFIG_X86_PMEM_LEGACY=y
CONFIG_VIRTIO_PMEM=m
CONFIG_BLK_DEV_PMEM=m
CONFIG_NVDIMM_DAX=y
CONFIG_DAX_DRIVER=y
CONFIG_DAX=y
CONFIG_DEV_DAX=m
CONFIG_DEV_DAX_PMEM=m
CONFIG_DEV_DAX_HMEM=m
CONFIG_DEV_DAX_KMEM=m
CONFIG_DEV_DAX_PMEM_COMPAT=m
CONFIG_FS_DAX=y
CONFIG_FS_DAX_PMD=y
CONFIG_ARCH_HAS_PMEM_API=y
root@LancerW:~#
```

输出如上图结果，表明目前 Ubuntu 版本支持 DAX 与 PMEM

2. 检查可用内存

```
root@LancerW:~# dmesg | grep e820
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
[    0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000100000-0x00000000dffeffff] usable
[    0.000000] BIOS-e820: [mem 0x00000000dfff0000-0x00000000dfffffff] ACPI data
[    0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000100000000-0x00000001a75fffff] usable
[    0.001253] e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
[    0.001255] e820: remove [mem 0x000a0000-0x000fffff] usable
[    0.490737] e820: reserve RAM buffer [mem 0x0009fc00-0x0009ffff]
[    0.490737] e820: reserve RAM buffer [mem 0xdfff0000-0xdfffffff]
```

3. 配置 8G 空间模拟持久化内存，输入指令 vim /etc/default/grub，在其中添加指令 GRUB_CMDLINE_LINUX="memmap=8G!4G"，表示从 4G 内存开始模拟持久化内存，模拟空间共计 8G。如下图。

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash memmap=8G!4G"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"
```

4. 更新 grub，输入指令 update-grub，如下图

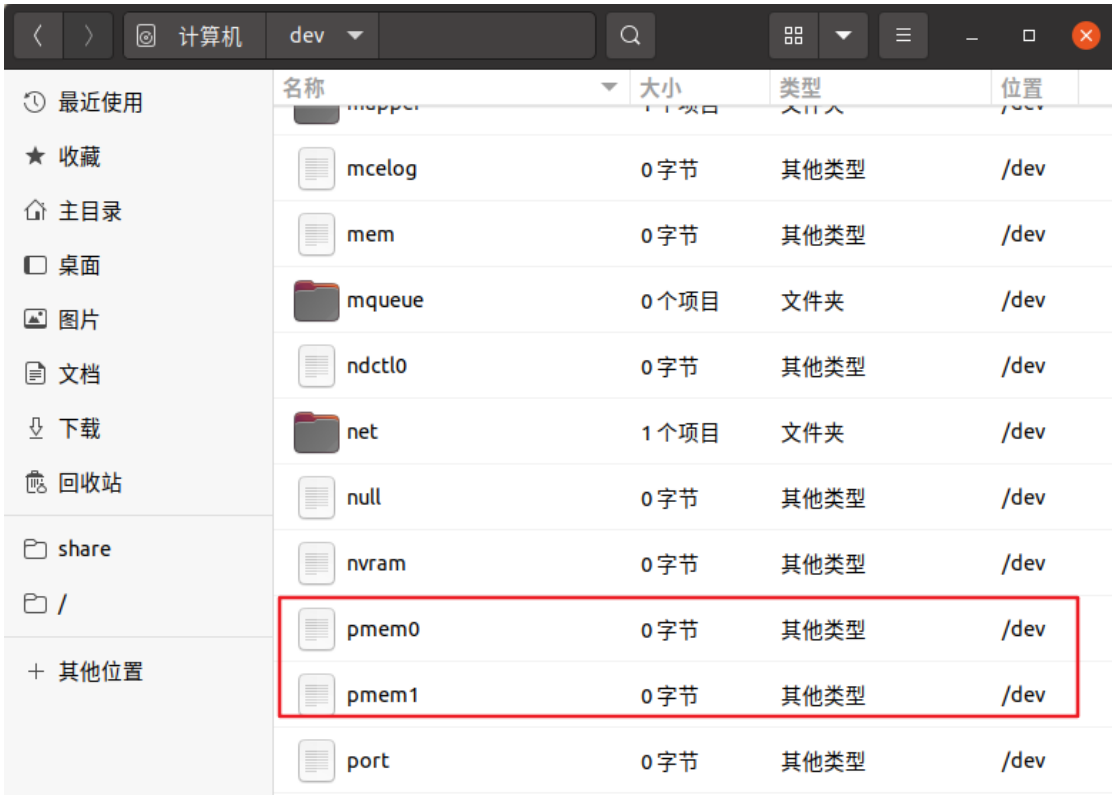
```
root@LancerW:~# update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
正在生成 grub 配置文件 ...
找到 Linux 镜像: /boot/vmlinuz-5.8.0-33-generic
找到 initrd 镜像: /boot/initrd.img-5.8.0-33-generic
找到 Linux 镜像: /boot/vmlinuz-5.8.0-31-generic
找到 initrd 镜像: /boot/initrd.img-5.8.0-31-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
完成
```

5. 查看是否配置成功

```
root@LancerW:~# dmesg | grep user
[    0.000000] user-defined physical RAM map:
[    0.000000] user: [mem 0x0000000000000000-0x000000000009fbff] usable
[    0.000000] user: [mem 0x000000000009fc00-0x000000000009ffff] reserved
[    0.000000] user: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[    0.000000] user: [mem 0x0000000000100000-0x00000000dffeffff] usable
[    0.000000] user: [mem 0x00000000dfff0000-0x00000000dfffffff] ACPI data
[    0.000000] user: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
[    0.000000] user: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
[    0.000000] user: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
[    0.000000] user: [mem 0x0000000100000000-0x00000001a75fffff] persistent (type 12)
[    0.000000] user: [mem 0x00000001a7600000-0x00000002ffffffff] persistent (type 12)
[    0.246378] Spectre V1 : Mitigation: usercopy/swapgs barriers and __user pointer sanitization
[    0.366320] thermal_sys: Registered thermal governor 'user_space'
[    2.081568] x86/mm: Checking user space page tables
[   10.303705] ppdev: user-space parallel port driver
```

可以看到状态为 persistent 的内存

在文件管理中可以找到 pmem0，pmem1 的设备，如图



6. 挂载文件系统，输入以下命令

1) mkdir /mnt/pmemdir

mnt 目录用于挂载，在其中新建 pmemdir 文件夹用于挂载持久化内存

2) mkfs.ext4 /dev/pmem0

用于格式化之前生成的持久化内存文件

3) mount -o dax /dev/pmem0 /mnt/pmemdir

将目录 pmemdir 挂载到持久化内存上，这样只需要操作文件夹就相当于操作持久化内存

结果如下图

二、 安装 PMDK 库

步骤如下

1. 安装 git

   apt install git

2. 安装 make

   apt install make

3. 克隆 pmdk 代码

   git clone https://github.com/pmem/pmdk.git

4. 安装依赖包

   apt install build-essential

   apt install libdaxctl-dev

   apt install libndctl-dev

   apt install pandoc

   apt install m4

   apt install libfabric-dev

5. 测试

   make test

   make check

6. 安装

   cd pmdk

   make install

最终截图如下

```
root@LancerW:~/pmdk# make
test -f .skip-doc || make -C doc all
make[1]: 进入目录“/root/pmdk/doc”
make[1]: 对“all”无需做任何事。
make[1]: 离开目录“/root/pmdk/doc”
make -C src all
make[1]: 进入目录“/root/pmdk/src”
make -C libpmem
make[2]: 进入目录“/root/pmdk/src/libpmem”
cc -MD -c -o ../nondebug/libpmem/alloc.o  -std=gnu99 -fno-common -pthread -DSRCV
ERSION=\"1.10+git70.g173f4aed5\" -DSDS_ENABLED -DNDCTL_ENABLED=1   -I../../src/.
./src/libpmem2/x86_64 -DAVX512F_AVAILABLE=1 -I. -I../../src/../src/libpmem2 -I..
/include -I../common/ -I../core/  -fPIC  ../../src/../src/core/alloc.c
../../src/../utils/check-os.sh ../../src/../utils/os-banned  ../nondebug/libpmem
/alloc.o  ../../src/../src/core/alloc.c
```

```
root@LancerW:~/pmdk# make install
test -f .skip-doc || make -C doc all
make[1]: 进入目录“/root/pmdk/doc”
make[1]: 对“all”无需做任何事。
make[1]: 离开目录“/root/pmdk/doc”
make -C src all
make[1]: 进入目录“/root/pmdk/src”
make -C libpmem
make[2]: 进入目录“/root/pmdk/src/libpmem”
make[2]: 对“all”无需做任何事。
make[2]: 离开目录“/root/pmdk/src/libpmem”
make -C libpmem DEBUG=1
make[2]: 进入目录“/root/pmdk/src/libpmem”
make[2]: 对“all”无需做任何事。
make[2]: 离开目录“/root/pmdk/src/libpmem”
make -C libpmemblk
make[2]: 进入目录“/root/pmdk/src/libpmemblk”
make[2]: 对“all”无需做任何事。
make[2]: 离开目录“/root/pmdk/src/libpmemblk”
```