

数据库课程设计环境配置报告

18341019 李怡臻

一、模拟 NVM 环境

1. 检查内核是否支持 DAX 和 PMEM

```
root@liyzh66-virtual-machine:/home/liyzh66# egrep '(DAX|PMEM)' /boot/config-5.4.0-56-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_KMEM=m
CONFIG_DEV_DAX_PMEM_COMPAT=m
CONFIG_FS_DAX=y
CONFIG_FS_DAX_PMD=y
CONFIG_ARCH_HAS_PMEM_API=y
```

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@liyzh66-virtual-machine:/home/liyzh66# dmesg | grep e820
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000009e7ff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000000009e800-0x00000000000009ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x000000000000dc000-0x000000000000ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000100000-0x00000000007fedffff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000007fee0000-0x00000000007fefffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x00000000007fef0000-0x00000000007fefffff] ACPI NVS
[ 0.000000] BIOS-e820: [mem 0x00000000007fff0000-0x00000000007fffffff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000f0000000-0x0000000000f7fffffff] reserved
[ 0.000000] BIOS-e820: [mem 0x000000000fec00000-0x000000000fec0ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fffe0000-0x00000000ffffffff] reserved
[ 0.004520] e820: update [mem 0x00000000-0x000000fff] usable ==> reserved
[ 0.004523] e820: remove [mem 0x000a0000-0x0000ffff] usable
[ 0.763280] e820: reserve RAM buffer [mem 0x0009e800-0x0009ffff]
[ 0.763282] e820: reserve RAM buffer [mem 0x7fee0000-0x7fffffff]
```

3. 配置 4G 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=4G!4G"`，表示从

4G 内存开始模拟持久化内存，模拟空间共计 4G。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"
GRUB_CMDLINE_LINUX="memmap=4G!4G"

# 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"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

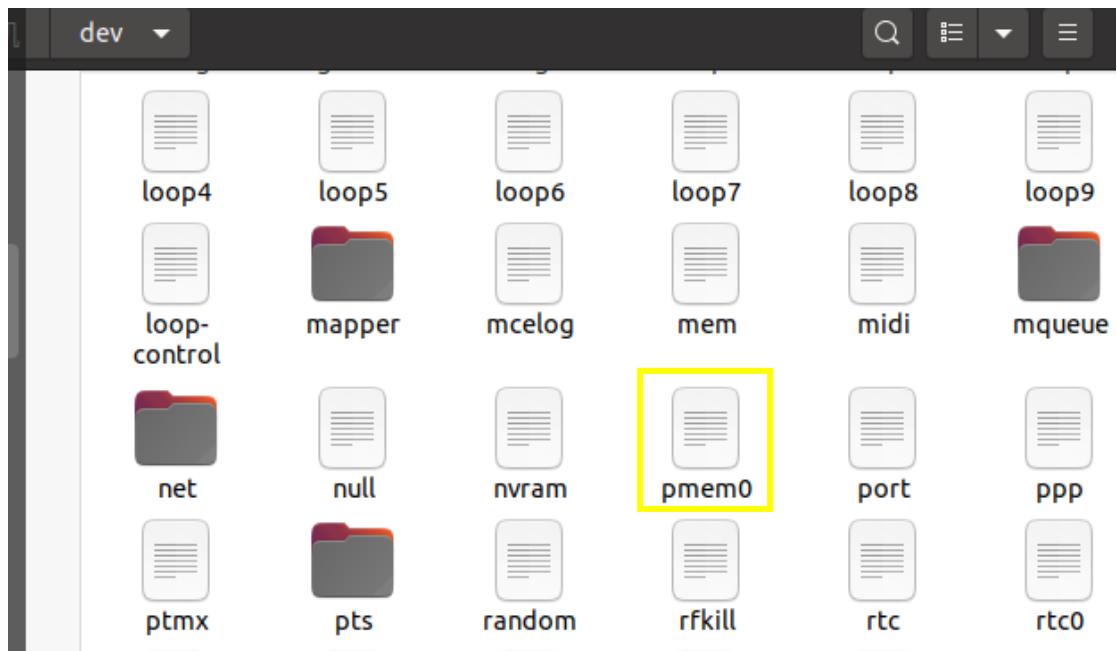
# Uncomment to disable generation of recovery mode menu entries
"/etc/default/grub" [只读] 33L, 1221C 1,1 顶端
```

4. 更新 grub, 输入指令 `update-grub`, 如下图

```
root@liyzh66-virtual-machine:/home/liyzh66# 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-5.4.0-58-generic
找到 initrd 镜像: /boot/initrd.img-5.4.0-58-generic
找到 Linux 镜像: /boot/vmlinuz-5.4.0-56-generic
找到 initrd 镜像: /boot/initrd.img-5.4.0-56-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
完成
```

5. 查看是否配置成功

可以看到状态为 `persistent` 的内存在文件管理中可以找到 `pmem0` 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

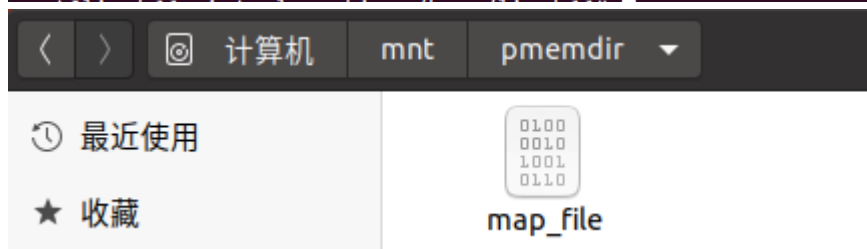
```

root@liyzh66-virtual-machine:/home/liyzh66# mkdir /mnt/pmemdir
mkdir: 无法创建目录 “/mnt/pmemdir”: 文件已存在
root@liyzh66-virtual-machine:/home/liyzh66# mkfs.ext4 /dev/pmem0
mke2fs 1.45.5 (07-Jan-2020)
创建含有 1048576 个块（每块 4k）和 262144 个inode的文件系统
文件系统UUID: 80ccd84c-9efc-41d9-aaf6-70ef63446b4b
超级块的备份存储于下列块:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

正在分配组表: 完成
正在写入inode表: 完成
创建日志 (16384 个块) 完成
写入超级块和文件系统账户统计信息: 已完成

root@liyzh66-virtual-machine:/home/liyzh66# mount -o dax /dev/pmem0/mnt/pmemdir
mount: /dev/pmem0/mnt/pmemdir: can't find in /etc/fstab.

```



二、安装 PMDK 库

步骤如下

1. 安装 git

apt install git

```

root@liyzh66-virtual-machine:/home/liyzh66# apt install git
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
git 已经是最新版 (1:2.25.1-1ubuntu3)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 203 个软件包未被升级。

```

2. 安装 make

apt install make

```

root@liyzh66-virtual-machine:/home/liyzh66# apt install make
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
make 已经是最新版 (4.2.1-1.2)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 203 个软件包未被升级。

```

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
```

```
root@liyzh66-virtual-machine:/home/liyzh66# apt install build-essential
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
build-essential 已经是最新版 (12.8ubuntu1.1)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 203 个软件包未被升级。
root@liyzh66-virtual-machine:/home/liyzh66# apt install libdaxctl-dev
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
libdaxctl-dev 已经是最新版 (67-1)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 203 个软件包未被升级。
root@liyzh66-virtual-machine:/home/liyzh66# apt install libndctl-dev
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
libndctl-dev 已经是最新版 (67-1)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 203 个软件包未被升级。
root@liyzh66-virtual-machine:/home/liyzh66# apt install pandoc
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
pandoc 已经是最新版 (2.5-3build2)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 203 个软件包未被升级。
root@liyzh66-virtual-machine:/home/liyzh66# apt install m4
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
m4 已经是最新版 (1.4.18-4)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 203 个软件包未被升级。
```

```
root@liyzh66-virtual-machine:/home/liyzh66# apt install libfabric-dev
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
libfabric-dev 已经是最新版 (1.6.2-3)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 203 个软件包未被升级。
```

5. 测试

```
make test
```

```
make check
```

6. 安装

cd pmdk

```
root@liyzh66-virtual-machine:/home/liyzh66# make test
make: *** 没有规则可制作目标“test”。 停止。
root@liyzh66-virtual-machine:/home/liyzh66# make check
make: *** 没有规则可制作目标“check”。 停止。
root@liyzh66-virtual-machine:/home/liyzh66# cd pmdk
```

make install 最终截图如下

```
root@liyzh66-virtual-machine:/home/liyzh66# make test
make: *** 没有规则可制作目标“test”。 停止。
root@liyzh66-virtual-machine:/home/liyzh66# make check
make: *** 没有规则可制作目标“check”。 停止。
root@liyzh66-virtual-machine:/home/liyzh66# cd pmdk
root@liyzh66-virtual-machine:/home/liyzh66/pmdk# make install
test -f .skip-doc || make -C doc all
make[1]: 进入目录“/home/liyzh66/pmdk/doc”
make[1]: 对“all”无需做任何事。
make[1]: 离开目录“/home/liyzh66/pmdk/doc”
make -C src all
make[1]: 进入目录“/home/liyzh66/pmdk/src”
make -C libpmem
make[2]: 进入目录“/home/liyzh66/pmdk/src/libpmem”
mkdir -p ../nondebug/libpmem
```