SNIA PERSISTENT MEMORY PM SUMMIT

JANUARY 24, 2019 | SANTA CLARA, CA

Part 1 - Has anyone seen my Persistent Memory?

Stephen Bates, CTO, Eideticom

Linux Support for Persistent Memory

Welcome!

First step: ssh host@35.239.91.230 (password = “pmrocks!”)

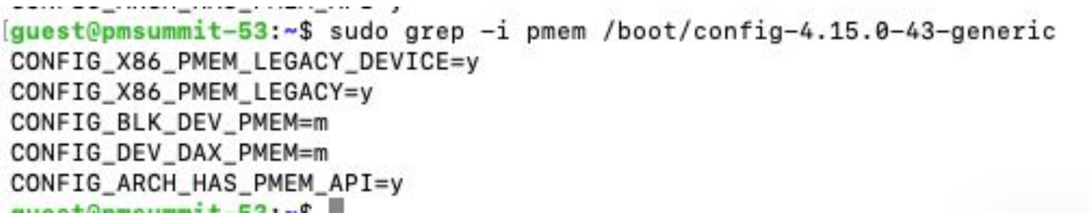
Let’s start with some logistics:

- Can you ssh into your VM? ssh guest@192.168.122.(100+X) where X is your pmsummit-X VM.
- sudo emacs etc/hostname && edit pmsummit-X (replace X with YOUR VM number).
- sudo apt update && sudo apt upgrade
- sudo reboot now
- **WARNING: Today’s hackathon is running on emulated PM. Ignore performance metrics….**
- **REQUEST: This is the first time using this framework. Feeback is welcome!**

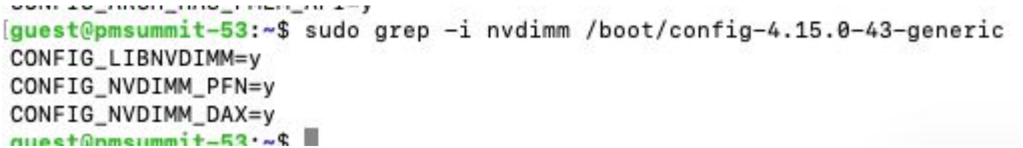
Is your Linux kernel PM aware?

Check your kernel config (/boot/config or /proc/config

```
CONFIG_BLK_DEV_RAM_DAX=y
CONFIG_FS_DAX=y
CONFIG_X86_PMEM_LEGACY=y
CONFIG_LIBNVDIMM=y
CONFIG_BLK_DEV_PMEM=m
CONFIG_ARCH_HAS_PMEM_API=y
CONFIG_TRANSPARENT_HUGEPAGE=y
CONFIG_MEMORY_HOTPLUG=y
CONFIG_MEMORY_HOTREMOVE=y
CONFIG_ZONE_DEVICE=y
CONFIG_FS_DAX_PMD=y
```



```
guest@pmsummit-53:~$ sudo grep -i pmem /boot/config-4.15.0-43-generic
CONFIG_X86_PMEM_LEGACY_DEVICE=y
CONFIG_X86_PMEM_LEGACY=y
CONFIG_BLK_DEV_PMEM=m
CONFIG_DEV_DAX_PMEM=m
CONFIG_ARCH_HAS_PMEM_API=y
guest@pmsummit-53:~$
```



```
guest@pmsummit-53:~$ sudo grep -i nvdimm /boot/config-4.15.0-43-generic
CONFIG_LIBNVDIMM=y
CONFIG_NVDIMM_PFN=y
CONFIG_NVDIMM_DAX=y
guest@pmsummit-53:~$
```

emacs drivers/nvdimm/Kconfig

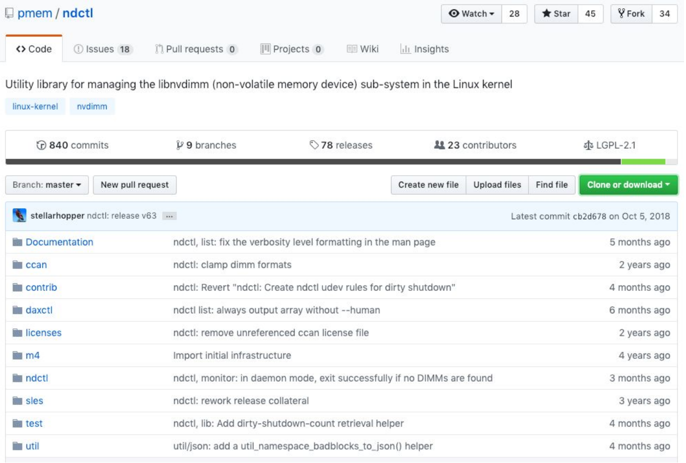
https://nvdimm.wiki.kernel.org/

The Linux kernel has great PM support but it has to be enabled. Most major Linux distributions are now doing this in the kernels they ship. E.g. our hackathon today uses Ubuntu 18.04 (aka Bionic Beaver) with a 4.15 based kernel.

Track development via linux-nvdimm mailing list!

Managing PM in Linux: **ndctl**

- How do you determine if you have PM in your system?
- How do you manage the PM in your system?
- How do you determine the health of PM in your system?

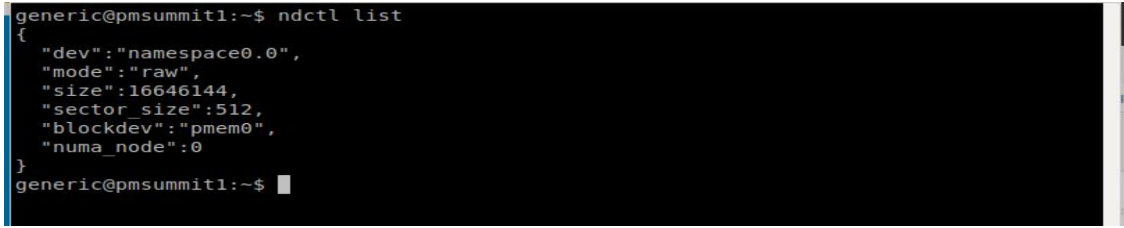


Utility library for managing the libnvdimm (non-volatile memory device) sub-system in the Linux kernel

stellarhopper ndctl: release v63 — Latest commit cb2d678 on Oct 5, 2018

Documentation	ndctl, list: fix the verbosity level formatting in the man page	5 months ago
ccan	ndctl: clamp dimm formats	2 years ago
contrib	ndctl: Revert "ndctl: Create ndctl udev rules for dirty shutdown"	4 months ago
daxctl	ndctl list: always output array without --human	6 months ago
licenses	ndctl: remove unreferenced ccan license file	2 years ago
m4	Import initial infrastructure	4 years ago
ndctl	ndctl, monitor: in daemon mode, exit successfully if no DIMMs are found	3 months ago
sles	ndctl: rework release collateral	3 years ago
test	ndctl, lib: Add dirty-shutdown-count retrieval helper	4 months ago
util	util/json: add a util_namespace_badblocks_to_json() helper	4 months ago

- Ties into physical layer specifications like NFIT and HMAT [1].

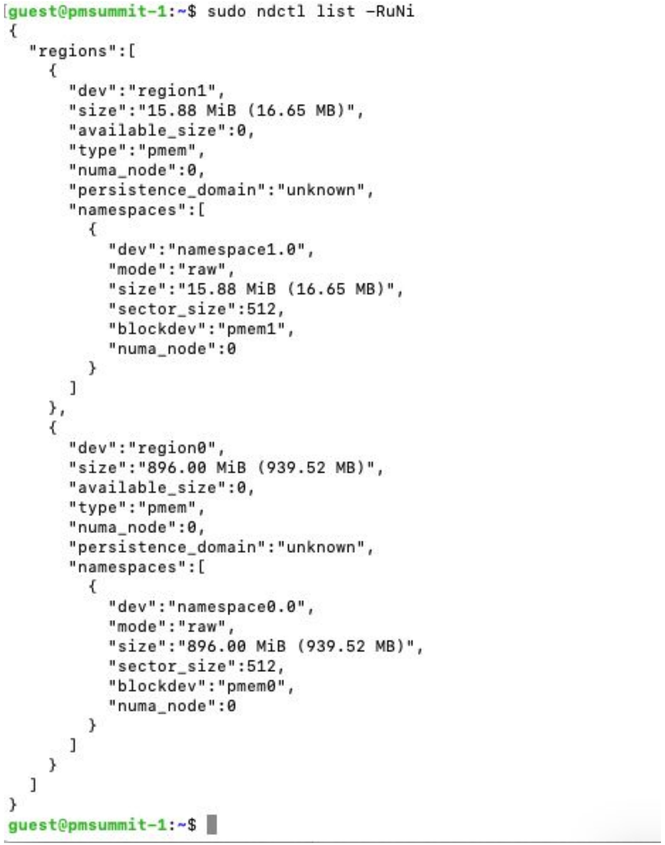


```
generic@pmsummit1:~$ ndctl list
{
  "dev":"namespace0.0",
  "mode":"raw",
  "size":16646144,
  "sector_size":512,
  "blockdev":"pmem0",
  "numa_node":0
}
generic@pmsummit1:~$ █
```

[1] http://www.uefi.org/sites/default/files/resources/ACPI_6_2.pdf

ndctl

- Try “sudo ndctl list -RuNi”
- You should see TWO NVDIMMs:
  - One is 16MiB or so.
  - One is 1GiB or so.
- Note both these NVDIMMs have labels are are in “raw” mode.

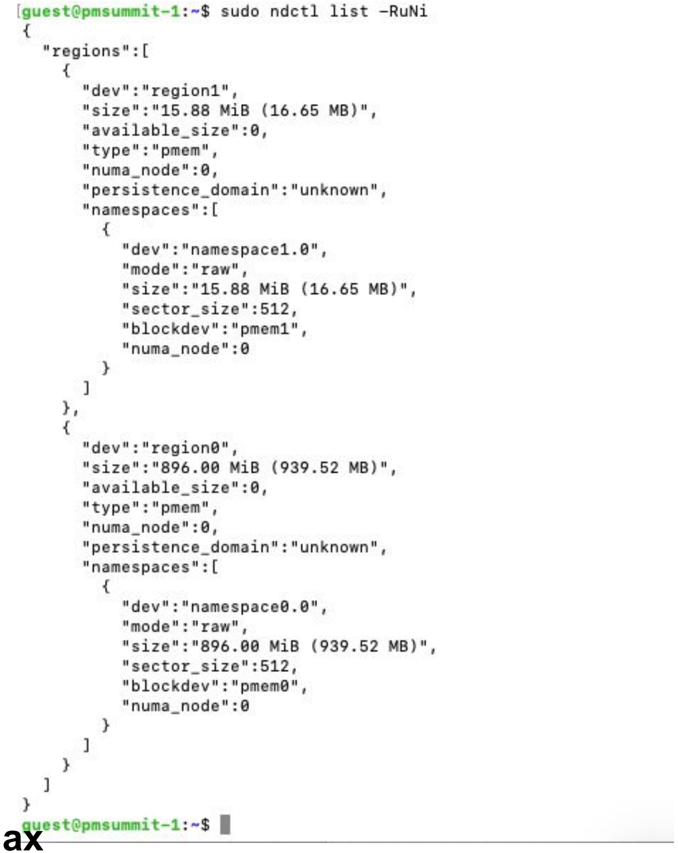


```
guest@pmsummit-1:~$ sudo ndctl list -RuNi
{
  "regions":[
    {
      "dev":"region1",
      "size":"15.88 MiB (16.65 MB)",
      "available_size":0,
      "type":"pmem",
      "numa_node":0,
      "persistence_domain":"unknown",
      "namespaces":[
        {
          "dev":"namespace1.0",
          "mode":"raw",
          "size":"15.88 MiB (16.65 MB)",
          "sector_size":512,
          "blockdev":"pmem1",
          "numa_node":0
        }
      ]
    },
    {
      "dev":"region0",
      "size":"896.00 MiB (939.52 MB)",
      "available_size":0,
      "type":"pmem",
      "numa_node":0,
      "persistence_domain":"unknown",
      "namespaces":[
        {
          "dev":"namespace0.0",
          "mode":"raw",
          "size":"896.00 MiB (939.52 MB)",
          "sector_size":512,
          "blockdev":"pmem0",
          "numa_node":0
        }
      ]
    }
  ]
}
guest@pmsummit-1:~$
```

NVDIMM Labels

- NVDIMM labels are optional on NVDIMMs.
- Store meta-data relevant to the NVDIMM
- Also allows us to divide the NVDIMM(s) into regions called namespaces.

https://pmem.io/documents/NVDIMM_Namespace_Spec.pdf



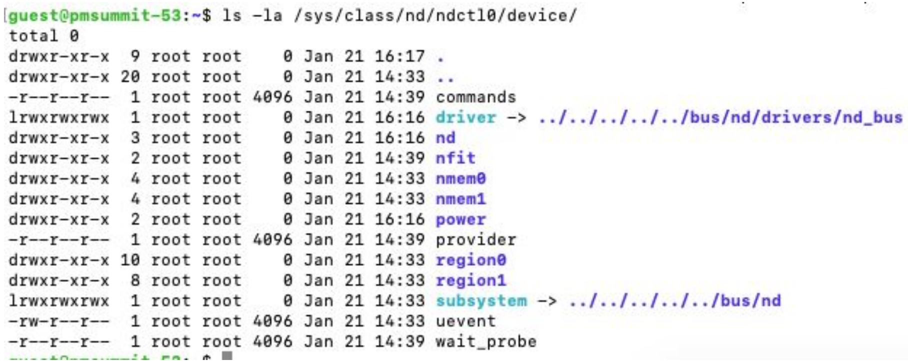
```
guest@pmsummit-1:~$ sudo ndctl list -RuNi
{
  "regions":[
    {
      "dev":"region1",
      "size":"15.88 MiB (16.65 MB)",
      "available_size":0,
      "type":"pmem",
      "numa_node":0,
      "persistence_domain":"unknown",
      "namespaces":[
        {
          "dev":"namespace1.0",
          "mode":"raw",
          "size":"15.88 MiB (16.65 MB)",
          "sector_size":512,
          "blockdev":"pmem1",
          "numa_node":0
        }
      ]
    },
    {
      "dev":"region0",
      "size":"896.00 MiB (939.52 MB)",
      "available_size":0,
      "type":"pmem",
      "numa_node":0,
      "persistence_domain":"unknown",
      "namespaces":[
        {
          "dev":"namespace0.0",
          "mode":"raw",
          "size":"896.00 MiB (939.52 MB)",
          "sector_size":512,
          "blockdev":"pmem0",
          "numa_node":0
        }
      ]
    }
  ]
}
guest@pmsummit-1:~$ 
```

sudo ndctl create-namespace -f -e namespace0.0 --mode fsdax

SNIA PERSISTENT MEMORY PM SUMMIT
JANUARY 24, 2019 | SANTA CLARA, CA

NVDIMM in sysfs

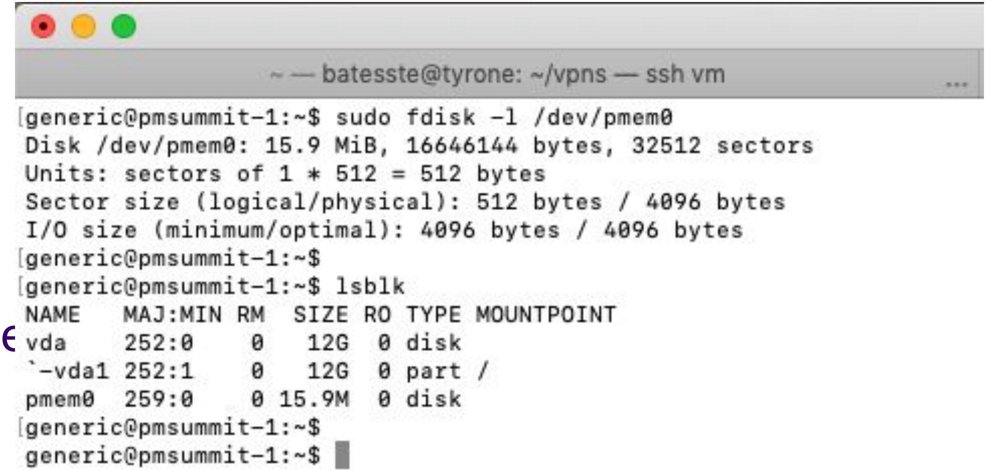
- sysfs is a kernel based pseudo-filesystem used to obtain information about your system.
- NVDIMM related info is contained in /sys/class/nd
- Let’s poke around there...



```
guest@pmsummit-53:~$ ls -la /sys/class/nd/ndctl0/device/
total 0
drwxr-xr-x  9 root root    0 Jan 21 16:17 .
drwxr-xr-x 20 root root    0 Jan 21 14:33 ..
-r--r--r--  1 root root 4096 Jan 21 14:39 commands
lrwxrwxrwx  1 root root    0 Jan 21 16:16 driver -> ../../../../../bus/nd/drivers/nd_bus
drwxr-xr-x  3 root root    0 Jan 21 16:16 nd
drwxr-xr-x  2 root root    0 Jan 21 14:39 nfit
drwxr-xr-x  4 root root    0 Jan 21 14:33 nmem0
drwxr-xr-x  4 root root    0 Jan 21 14:33 nmem1
drwxr-xr-x  2 root root    0 Jan 21 16:16 power
-r--r--r--  1 root root 4096 Jan 21 14:39 provider
drwxr-xr-x 10 root root    0 Jan 21 14:33 region0
drwxr-xr-x  8 root root    0 Jan 21 14:33 region1
lrwxrwxrwx  1 root root    0 Jan 21 14:33 subsystem -> ../../../../../bus/nd
-rw-r--r--  1 root root 4096 Jan 21 14:33 uevent
-r--r--r--  1 root root 4096 Jan 21 14:39 wait_probe
```

Using PM in Linux: **A block device**

- PM can be consumed by Linux applications using a block interface.
- Useful when a file-system is not required.
- When possible the IO will optimize for PM (via DAX).
- All your favourite block-device tools can be used.



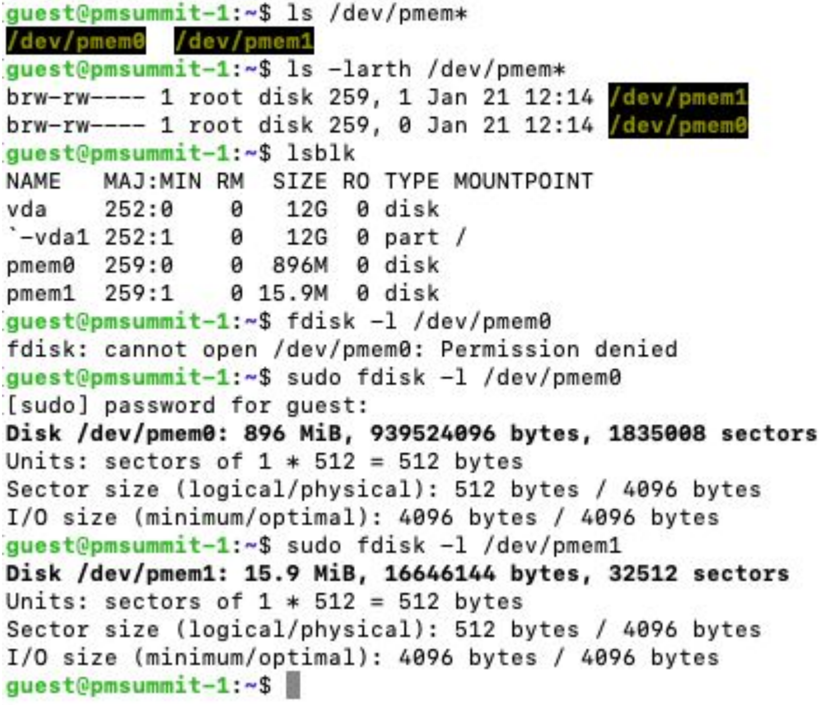
```
[generic@pmsummit-1:~$ sudo fdisk -l /dev/pmem0
 Disk /dev/pmem0: 15.9 MiB, 16646144 bytes, 32512 sectors
 Units: sectors of 1 * 512 = 512 bytes
 Sector size (logical/physical): 512 bytes / 4096 bytes
 I/O size (minimum/optimal): 4096 bytes / 4096 bytes
[generic@pmsummit-1:~$
[generic@pmsummit-1:~$ lsblk
 NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
 vda     252:0    0   12G  0 disk
 `-vda1  252:1    0   12G  0 part /
 pmem0   259:0    0 15.9M  0 disk
[generic@pmsummit-1:~$
 generic@pmsummit-1:~$ 
```

[1] http://www.uefi.org/sites/default/files/resources/ACPI_6_2.pdf

/dev/pmemX

sudo dd if=/dev/urandom of=/dev/pmem0 bs=32k count=16k

- Each NVDIMM is exposed as a pmem block device by our kernel.
- You can do all the normal block device things with it:
  - lsblk
  - fdisk -l
- We can also use dd, fio or a similar tool to write and read to the /dev/pmemX block device. Just like you would a HDD or SSD.

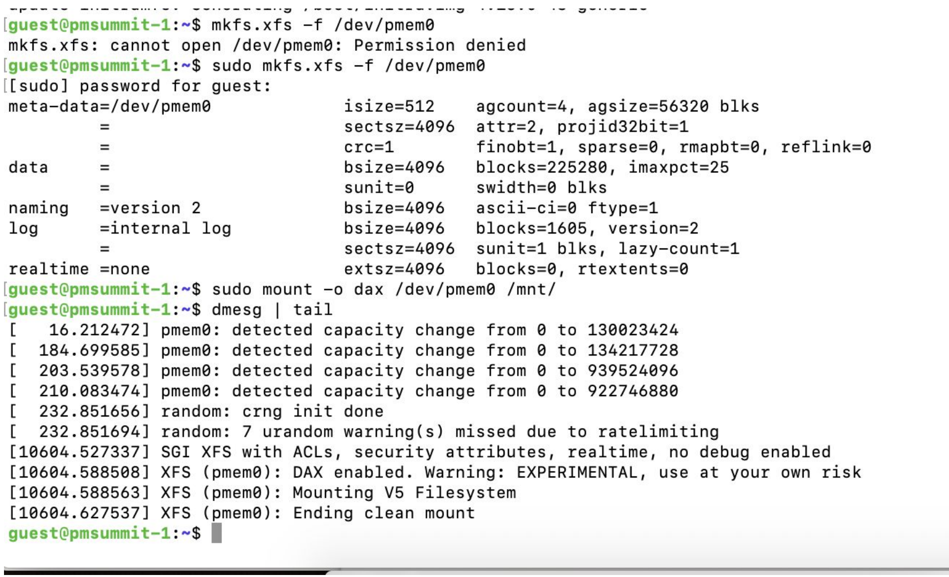


```
guest@pmsummit-1:~$ ls /dev/pmem*
/dev/pmem0  /dev/pmem1
guest@pmsummit-1:~$ ls -larth /dev/pmem*
brw-rw---- 1 root disk 259, 1 Jan 21 12:14 /dev/pmem1
brw-rw---- 1 root disk 259, 0 Jan 21 12:14 /dev/pmem0
guest@pmsummit-1:~$ lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda     252:0    0   12G  0 disk
`-vda1 252:1    0   12G  0 part /
pmem0   259:0    0  896M  0 disk
pmem1   259:1    0 15.9M  0 disk
guest@pmsummit-1:~$ fdisk -l /dev/pmem0
fdisk: cannot open /dev/pmem0: Permission denied
guest@pmsummit-1:~$ sudo fdisk -l /dev/pmem0
[sudo] password for guest:
Disk /dev/pmem0: 896 MiB, 939524096 bytes, 1835008 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
guest@pmsummit-1:~$ sudo fdisk -l /dev/pmem1
Disk /dev/pmem1: 15.9 MiB, 16646144 bytes, 32512 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
guest@pmsummit-1:~$
```

A PMEM device could be used as a fast block device in database or storage applications (e.g. the write journal in ZFS or RocksDB??)

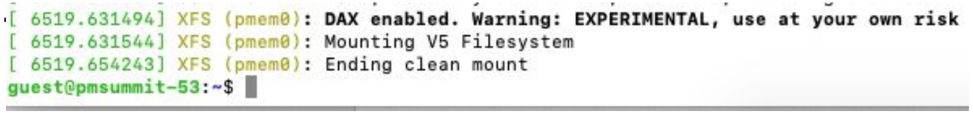
Using PM in Linux: **A filesystem**

- Like any standard block device we can put a filesystem over it.
- In this case we can put ANY filesystem we like on top.
- But if we use a DAX aware filesystem we get added benefits (PM optimizations).
- Now we can have files and directories and all that good stuff!
- In Linux both EXT4 and XFS have DAX support.



```
[guest@pmsummit-1:~$ mkfs.xfs -f /dev/pmem0
mkfs.xfs: cannot open /dev/pmem0: Permission denied
[guest@pmsummit-1:~$ sudo mkfs.xfs -f /dev/pmem0
[[sudo] password for guest:
meta-data=/dev/pmem0             isize=512    agcount=4, agsize=56320 blks
         =                       sectsz=4096  attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=0, rmapbt=0, reflink=0
data     =                       bsize=4096   blocks=225280, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=1605, version=2
         =                       sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[guest@pmsummit-1:~$ sudo mount -o dax /dev/pmem0 /mnt/
[guest@pmsummit-1:~$ dmesg | tail
[   16.212472] pmem0: detected capacity change from 0 to 130023424
[  184.699585] pmem0: detected capacity change from 0 to 134217728
[  203.539578] pmem0: detected capacity change from 0 to 939524096
[  210.083474] pmem0: detected capacity change from 0 to 922746880
[  232.851656] random: crng init done
[  232.851694] random: 7 urandom warning(s) missed due to ratelimiting
[10604.527337] SGI XFS with ACLs, security attributes, realtime, no debug enabled
[10604.588508] XFS (pmem0): DAX enabled. Warning: EXPERIMENTAL, use at your own risk
[10604.588563] XFS (pmem0): Mounting V5 Filesystem
[10604.627537] XFS (pmem0): Ending clean mount
guest@pmsummit-1:~$
```

XFS DAX



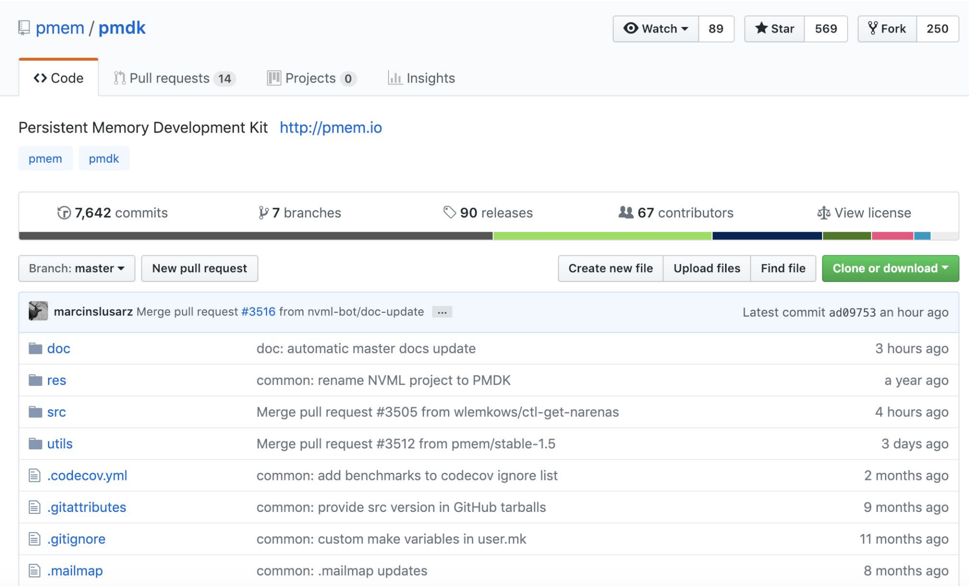
```
[ 6519.631494] XFS (pmem0): DAX enabled. Warning: EXPERIMENTAL, use at your own risk
[ 6519.631544] XFS (pmem0): Mounting V5 Filesystem
[ 6519.654243] XFS (pmem0): Ending clean mount
guest@pmsummit-53:~$
```

- Let’s put a DAX capable NVDIMMs.
  - sudo mkdir -p /mnt/pmem-fsdax
  - sudo apt install xfsprogs
  - sudo mkfs.xfs /dev/pmem0
  - sudo mount -o dax /dev/pmem0 /mnt/pmem-fsdax
- Check dmesg
- Check “sudo mount”
- Check “lsblk”

https://www.kernel.org/doc/Documentation/filesystems/dax.txt

Using PM in Linux: **mmap()**

- Now we have files that live in a PM-aware filesystem on a PM-aware block device on physical PM.
- mmap() has been around for a while ;-). Maps a file into the virtual address space of the running process.
- Now the world becomes our oyster (see PMDK for more!)



Persistent Memory Development Kit http://pmem.io

7,642 commits | 7 branches | 90 releases | 67 contributors | View license

marcinslusarz Merge pull request #3516 from nvml-bot/doc-update … Latest commit ad09753 an hour ago

doc	doc: automatic master docs update	3 hours ago
res	common: rename NVML project to PMDK	a year ago
src	Merge pull request #3505 from wlemkows/ctl-get-narenas	4 hours ago
utils	Merge pull request #3512 from pmem/stable-1.5	3 days ago
.codecov.yml	common: add benchmarks to codecov ignore list	2 months ago
.gitattributes	common: provide src version in GitHub tarballs	9 months ago
.gitignore	common: custom make variables in user.mk	11 months ago
.mailmap	common: .mailmap updates	8 months ago

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

Linux Support for Persistent Memory