FUJITSU

shaping tomorrow with you

The principle and enhancement of per-file DAX

Oct. 24 2020

Xiao Yang / Hao Li

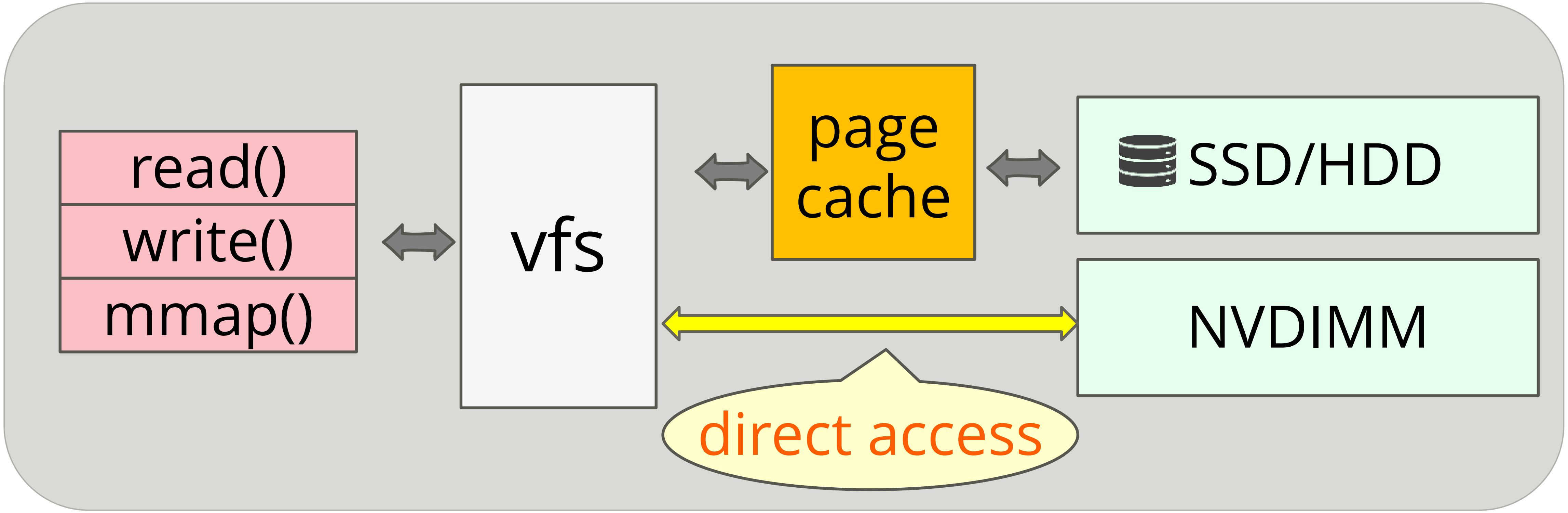
Agenda

- The Principle of per-file DAX
- The Enhancement of per-file DAX

The Principle of per-file DAX

Overview of DAX

- What is DAX?
  - Direct access
    - Copy data directly between pmem device and apps.
    - Bypass page cache.



Overview of per-file DAX

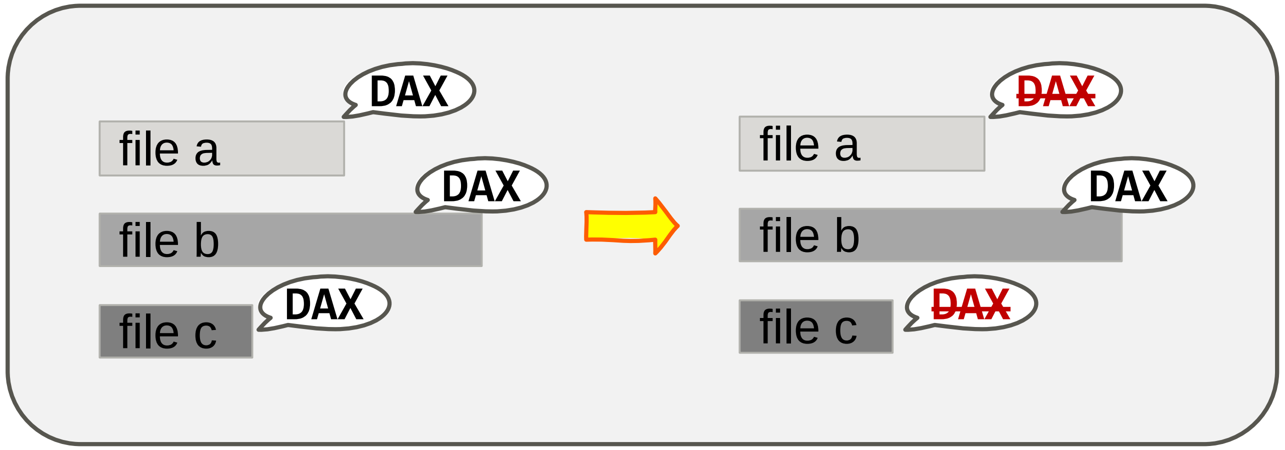
Use case for per-file DAX

Users only want to enable DAX on some specific files.

- Write operation on NVDIMM is a bit slower than on RAM.
- In another word, DAX write may slower than buffered write in some cases.

What is per-file DAX

Enable/Disable DAX for individual files.



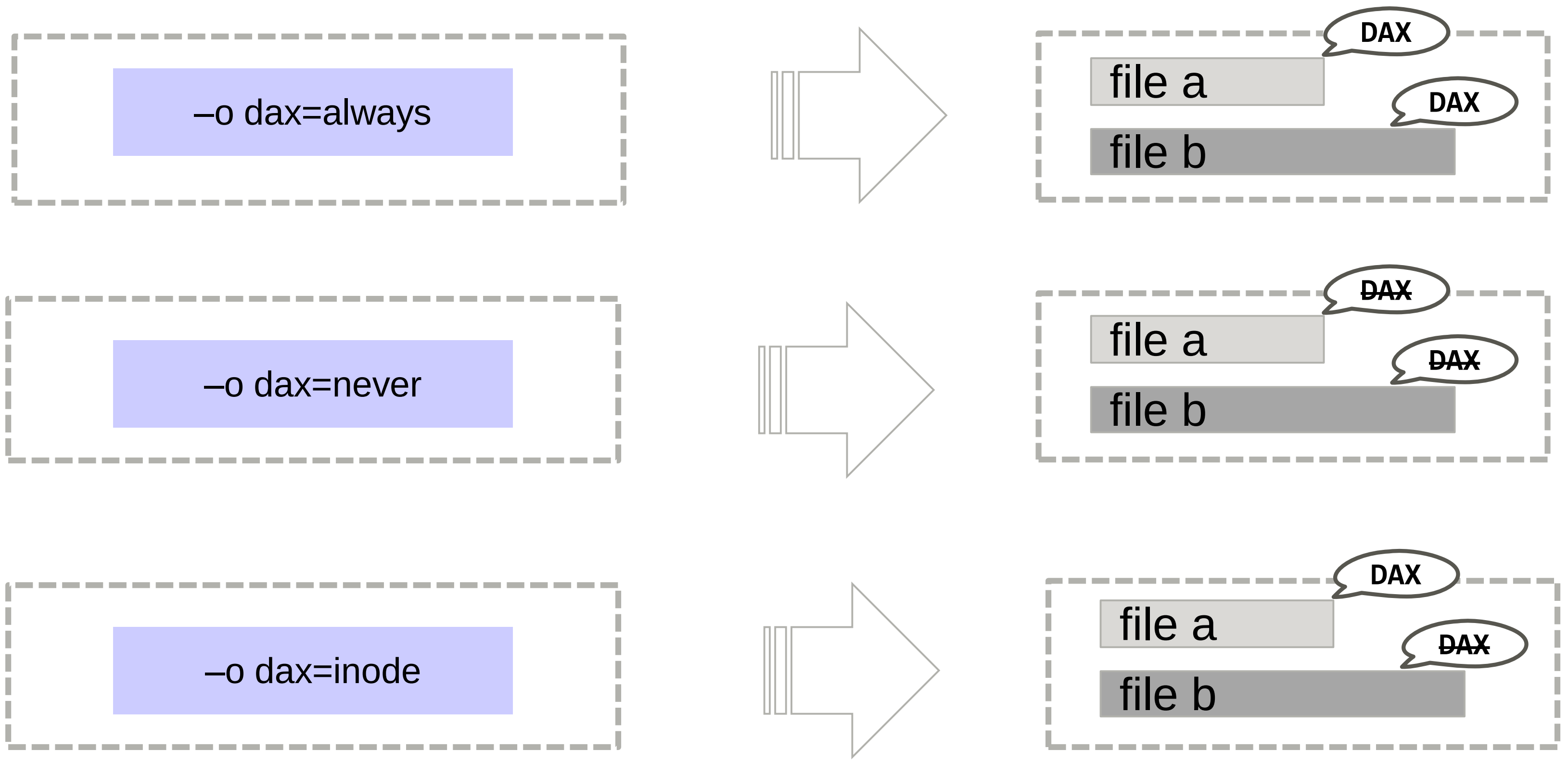
References

EXT4: https://lkml.org/lkml/2020/5/28/949

XFS: https://lkml.org/lkml/2020/4/27/1336

Introduction of dax mount options

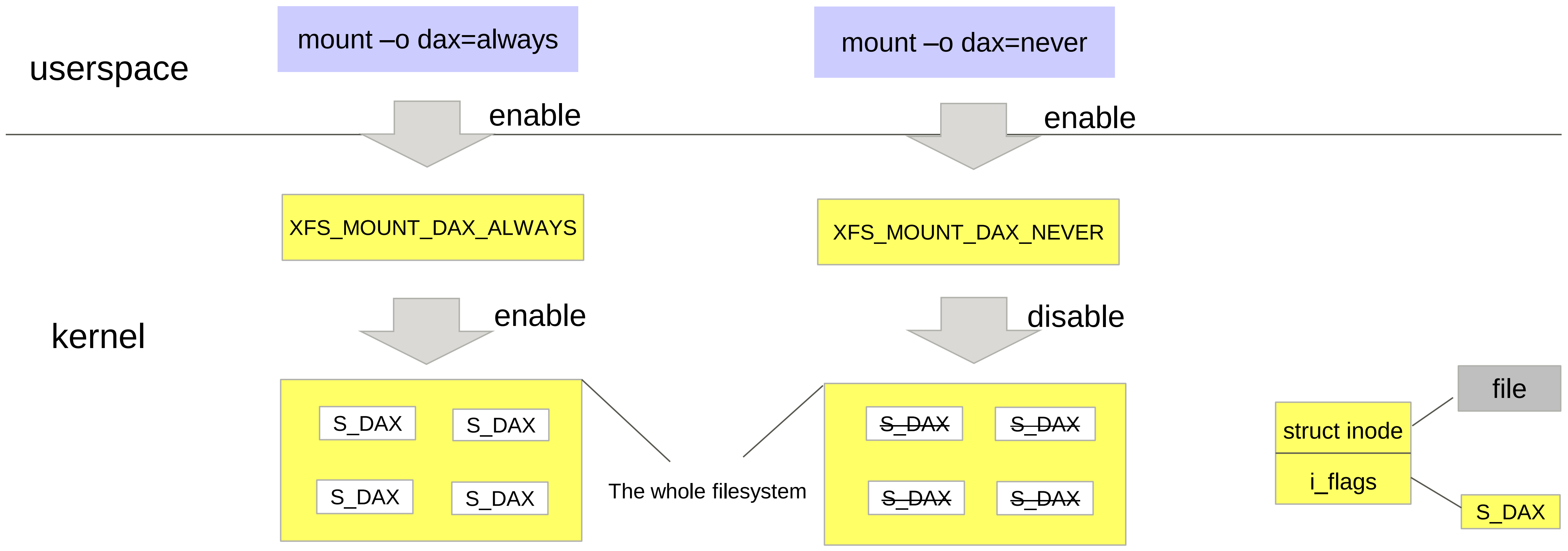
- Per-file DAX implements a tri-state dax mount options
  - -o dax=always/never controls DAX for all file in the whole filesystem
  - -o dax=inode controls DAX for individual files



How to control DAX?

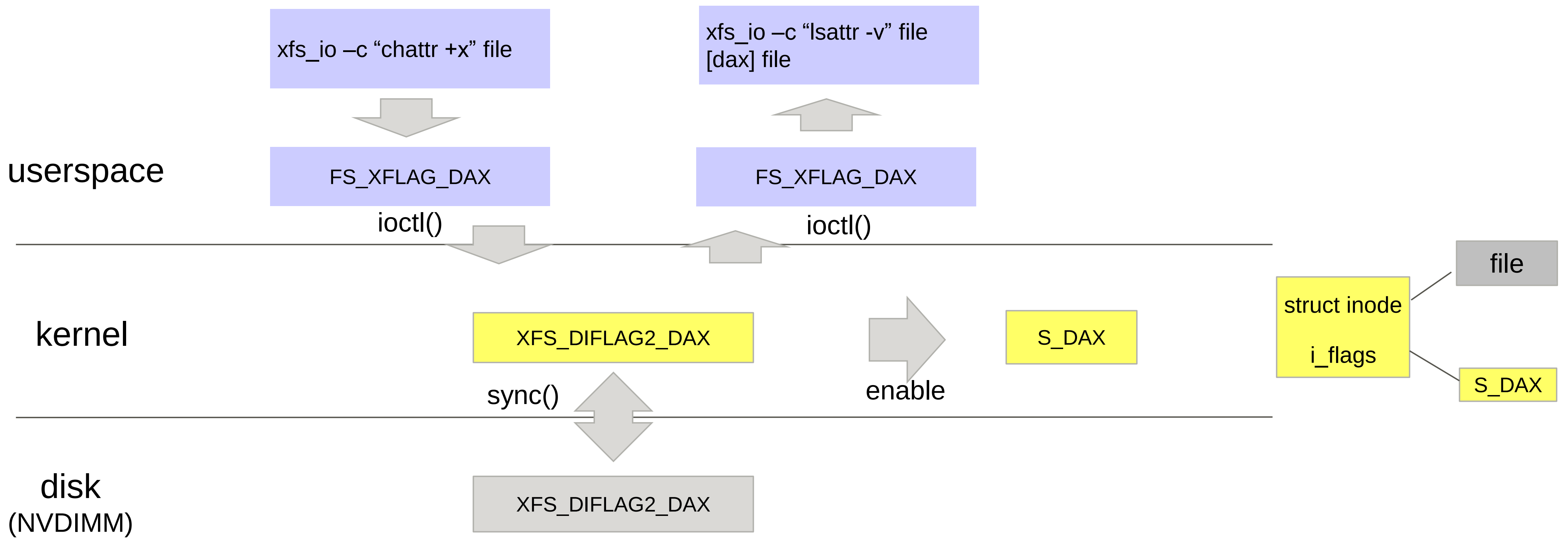
Control DAX by -o dax=always/never

- Introduction of three DAX flags
  - -o dax=always/never enables XFS_MOUNT_DAX_ALWAYS/NEVER
  - XFS_MOUNT_DAX_ALWAYS/NEVER enables/disables S_DAX which controls DAX operation



Control DAX by -o dax=inode

- Introduction of three DAX flags
  - XFS_DIFLAG2_DAX is a persistent flag on per-file
  - FS_XFLAG_DAX is used to set/get XFS_DIFLAG2_DAX
  - XFS_DIFLAG2_DAX enables S_DAX which controls DAX operation



Process of doing DAX operation

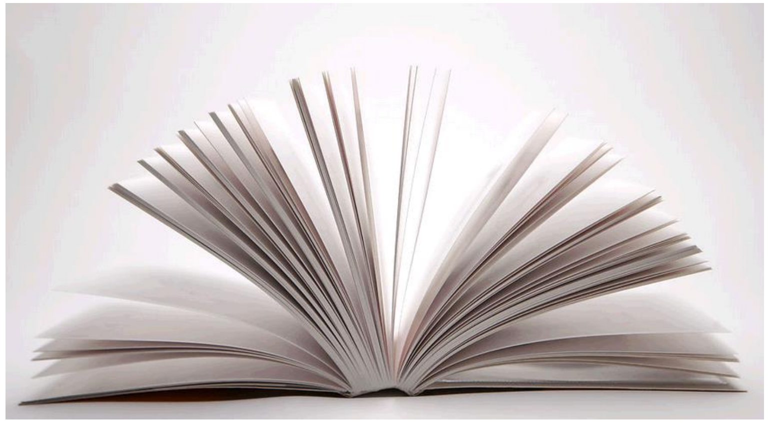
Process A(normal read/write)

```
read()/pread()
     -> vfs_read()                 check S_DAX
         -> xfs_file_read_iter()
             ->xfs_file_dax_read()
                 -> dax_iomap_rw()
                     -> …
```

```
write()/pwrite()                   check S_DAX
     -> vfs_write()
         -> xfs_file_write_iter()
           -> xfs_file_dax_write()
                 -> dax_iomap_rw()
                     -> …
```

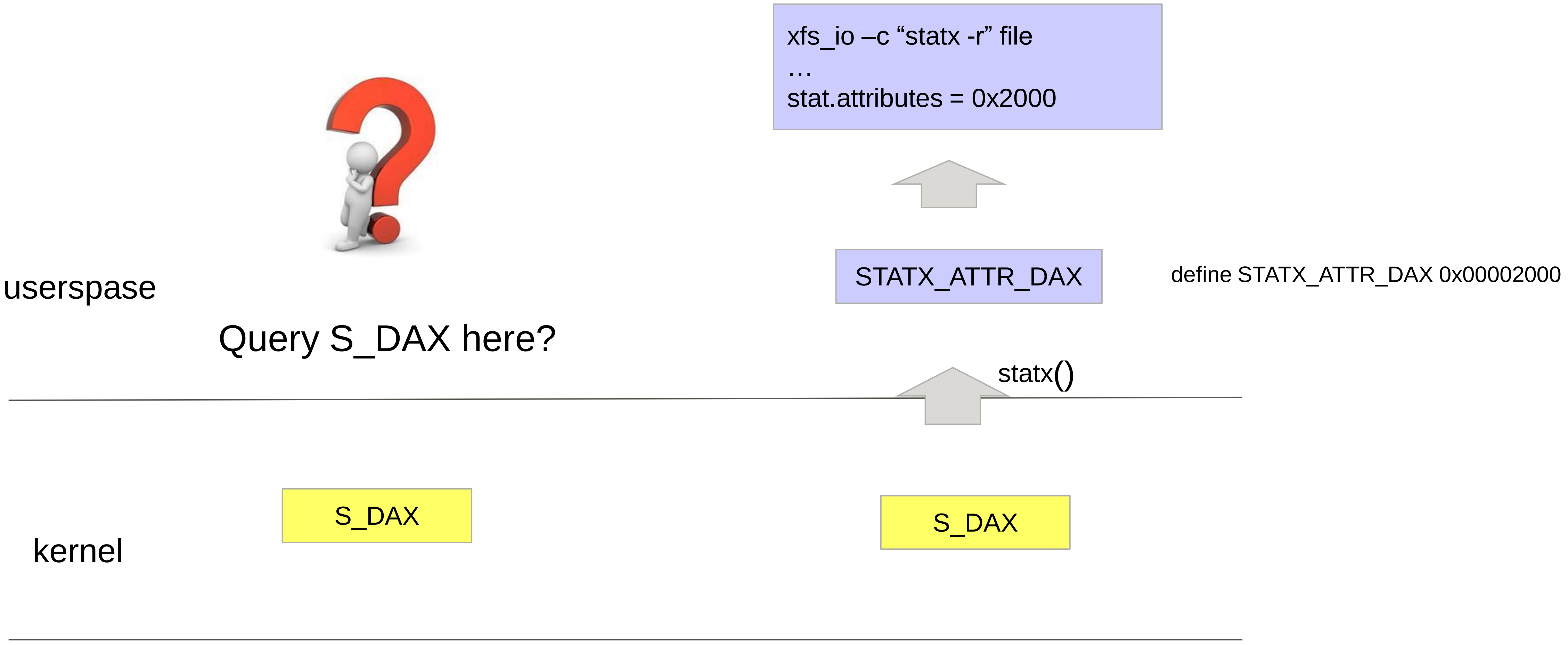
Process B(file mapping)

```
mmap()
     -> vm_mmap_pgoff()
         -> do_mmap()
              -> xfs_file_mmap()          check S_DAX
                 -> xfs_filemap_fault()
                    -> dax_iomap_fault()
                          ->…
```



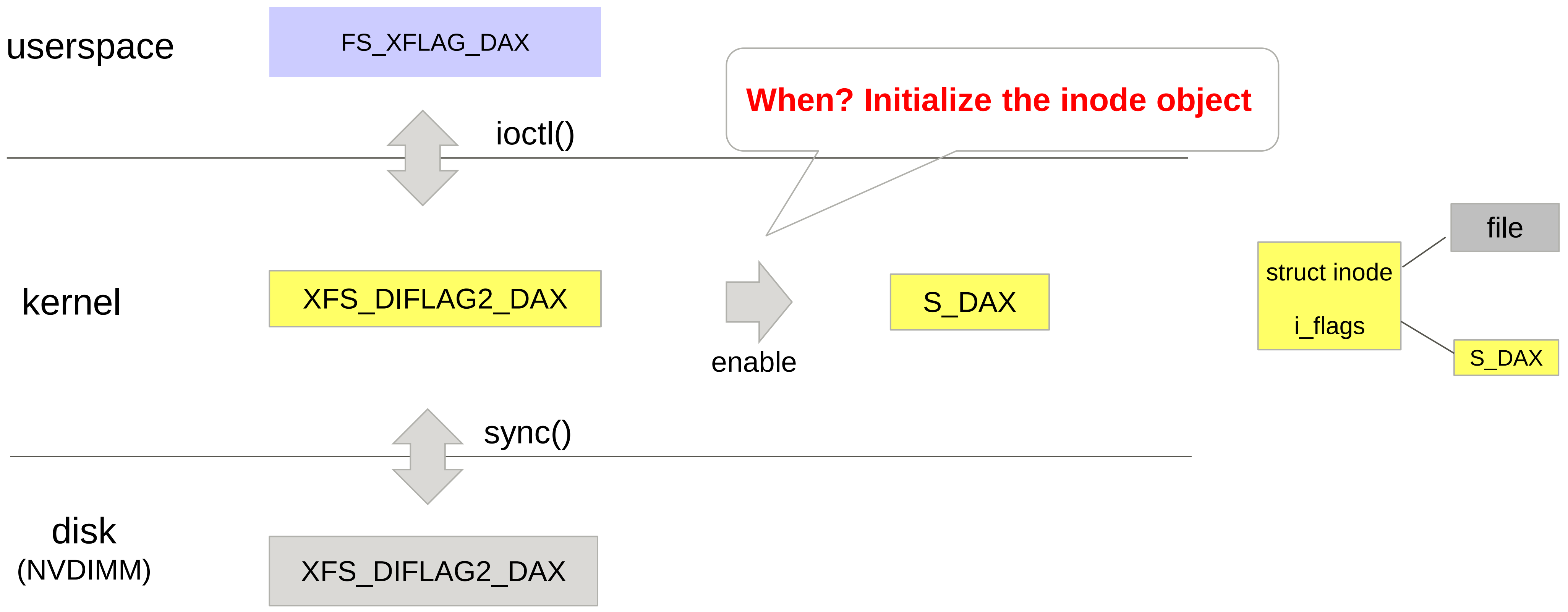
Query the state of S_DAX by statx(STATX_ATTR_DAX)

- Per-file DAX implements STATX_ATTR_DAX to query S_DAX.



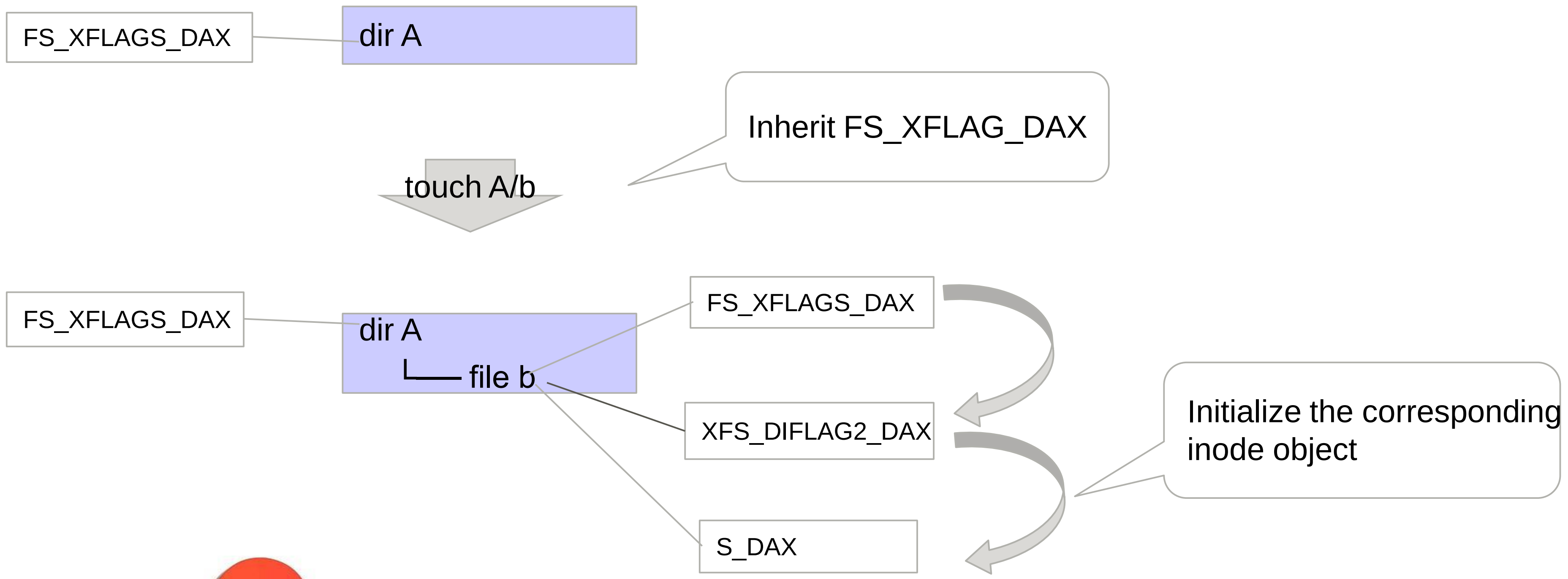
```
xfs_io –c “statx -r” file
…
stat.attributes = 0x2000
```

When to enable S_DAX by FS_XFLAG_DAX?



Method1: Inherit FS_XFLAG_DAX

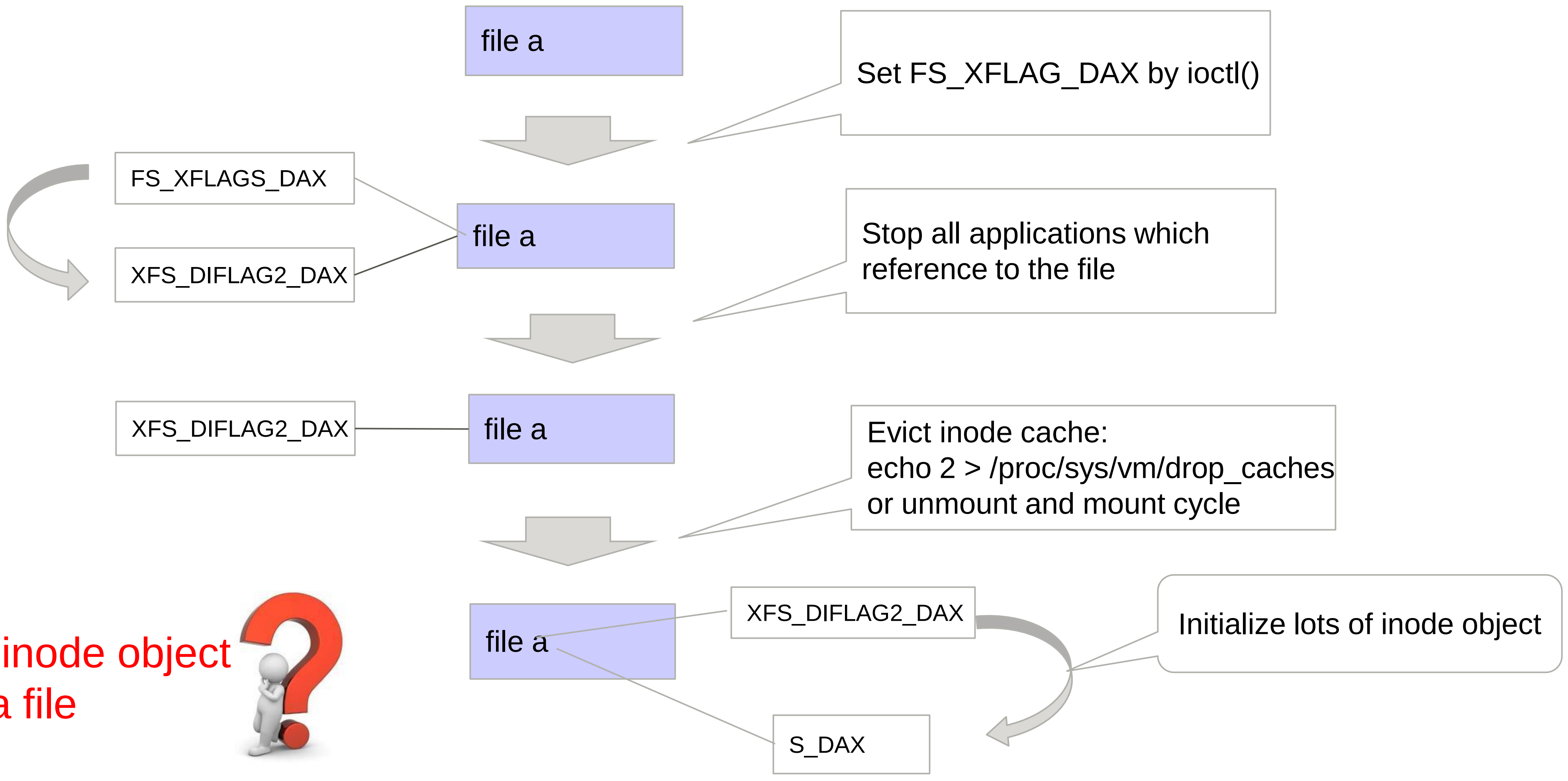
- Create a file under an existing directory with FS_XFLAG_DAX.



existing files

Method2: Evict inode cache

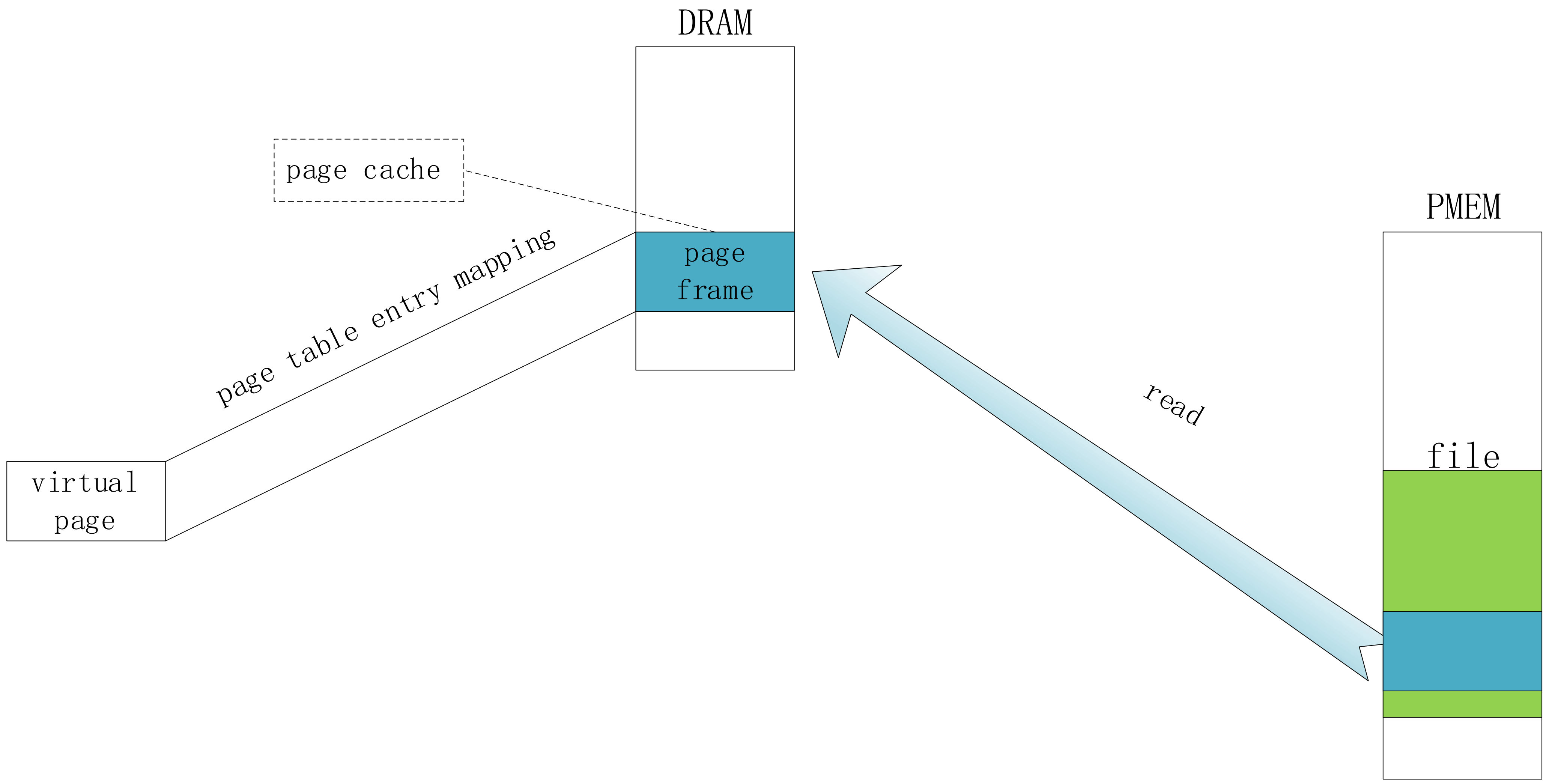
- Change FS_XFLAG_DAX on an existing file.


file a
Set FS_XFLAG_DAX by ioctl()
FS_XFLAGS_DAX
XFS_DIFLAG2_DAX
file a
Stop all applications which reference to the file
XFS_DIFLAG2_DAX
file a
Evict inode cache:
echo 2 > /proc/sys/vm/drop_caches
or unmount and mount cycle
file a
XFS_DIFLAG2_DAX
S_DAX
Initialize lots of inode object

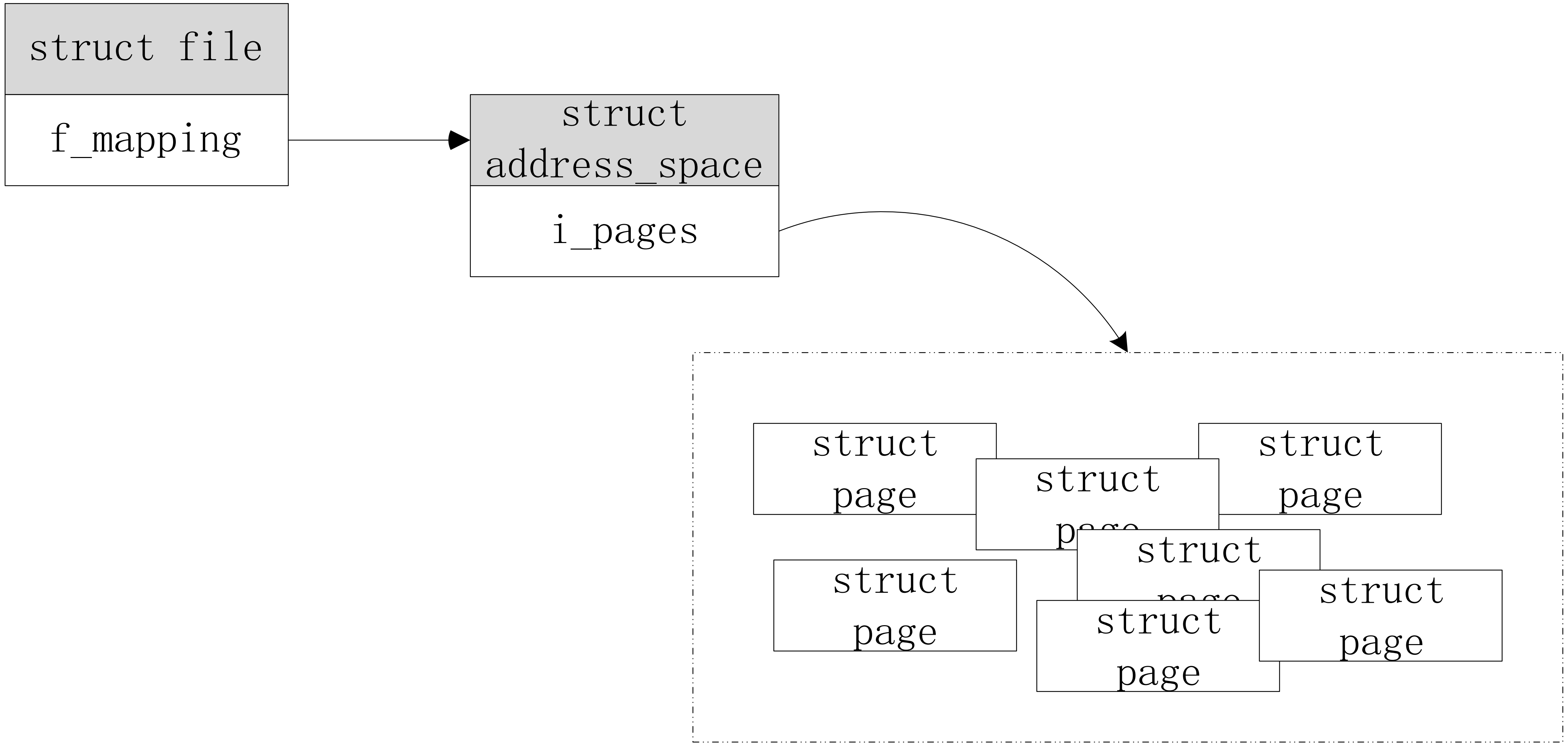
Drop lots of inode object for a file

The enhancement of per-file DAX

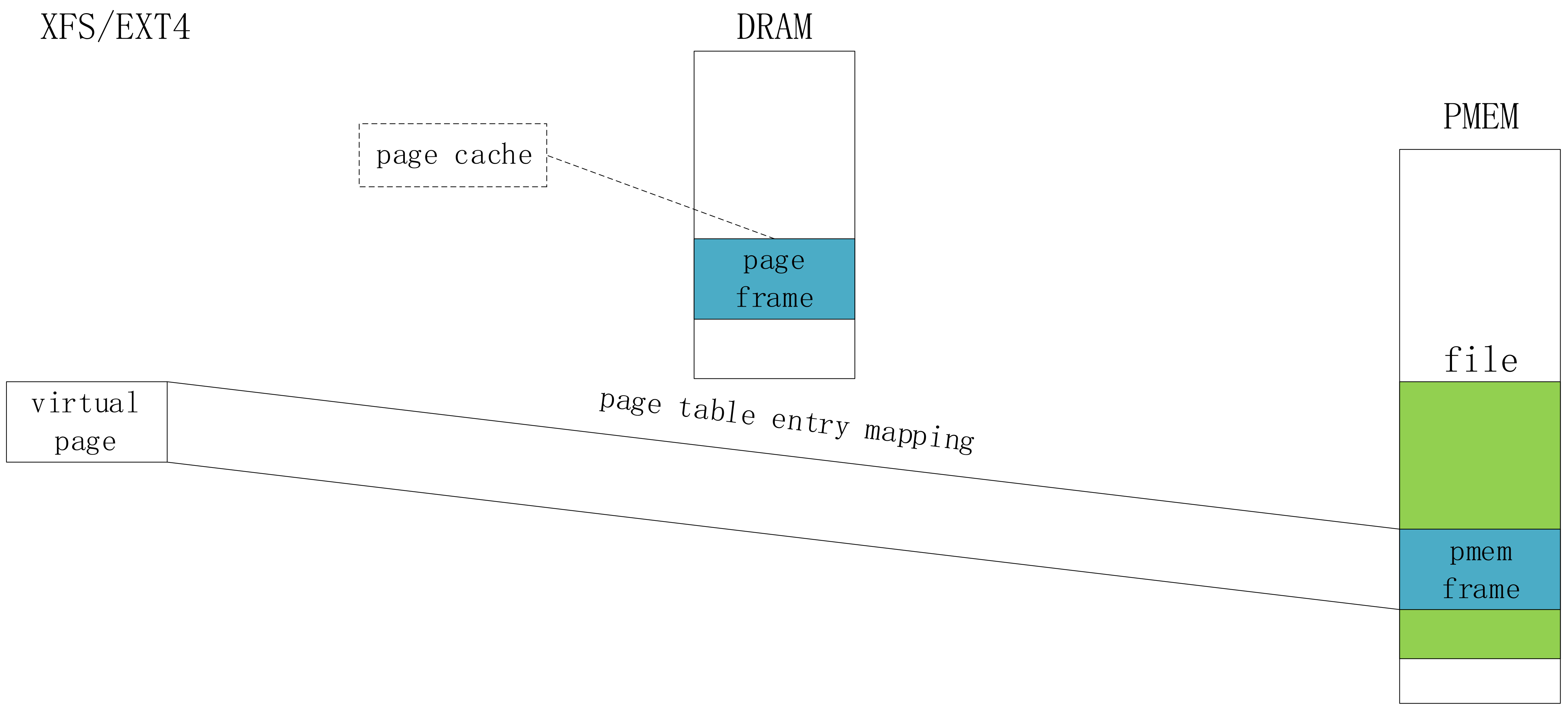
Non-DAX mode: PMEM with Page Cache



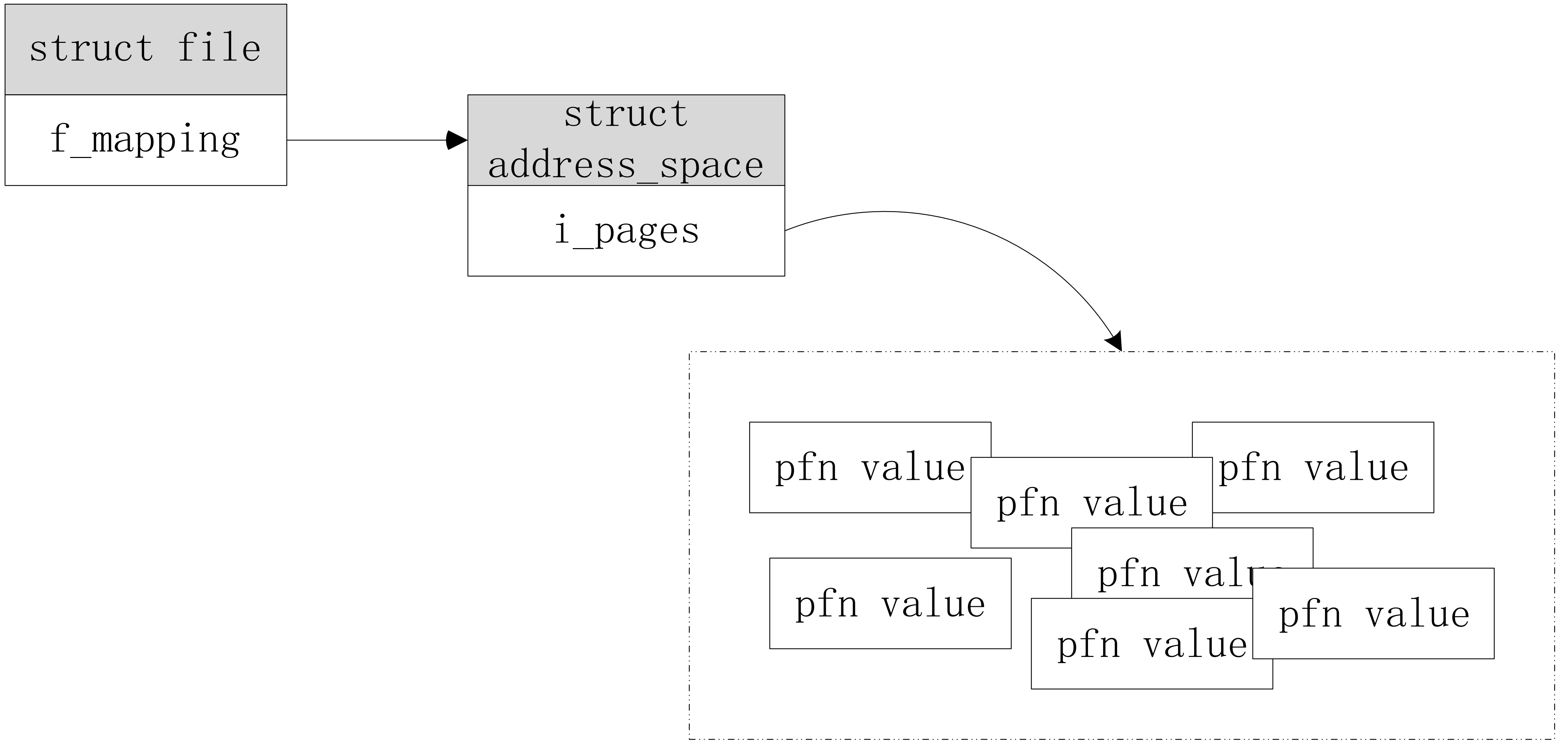
The radix tree in Non-DAX mode



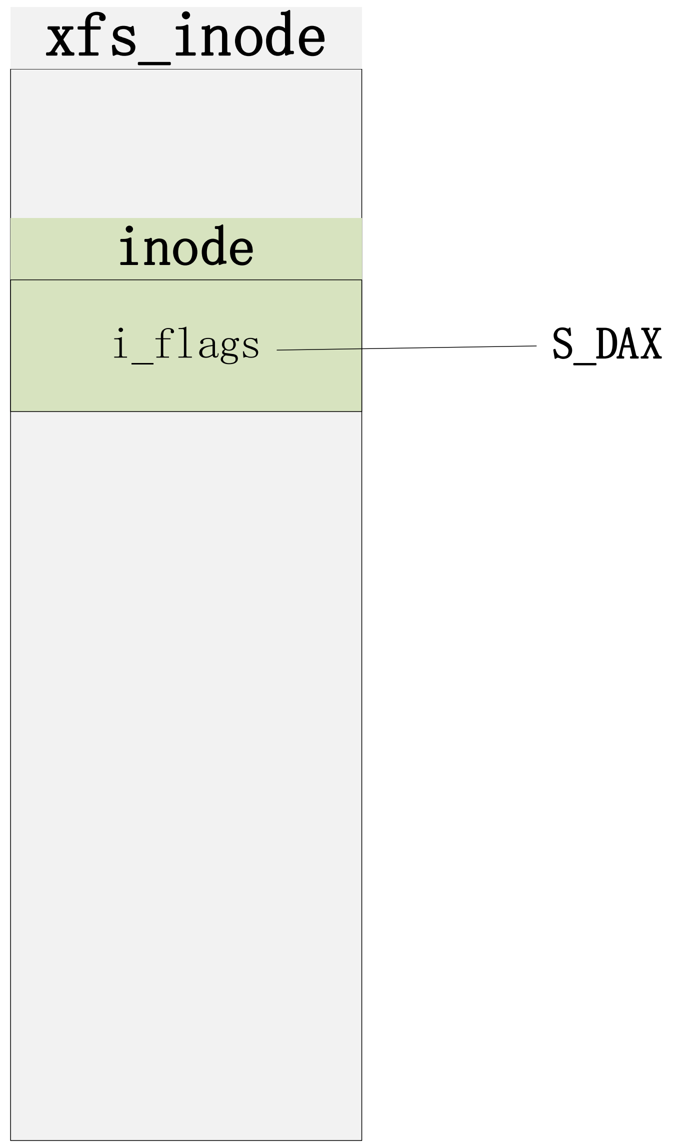
DAX mode: PMEM by-pass Page Cache



The radix tree in DAX mode



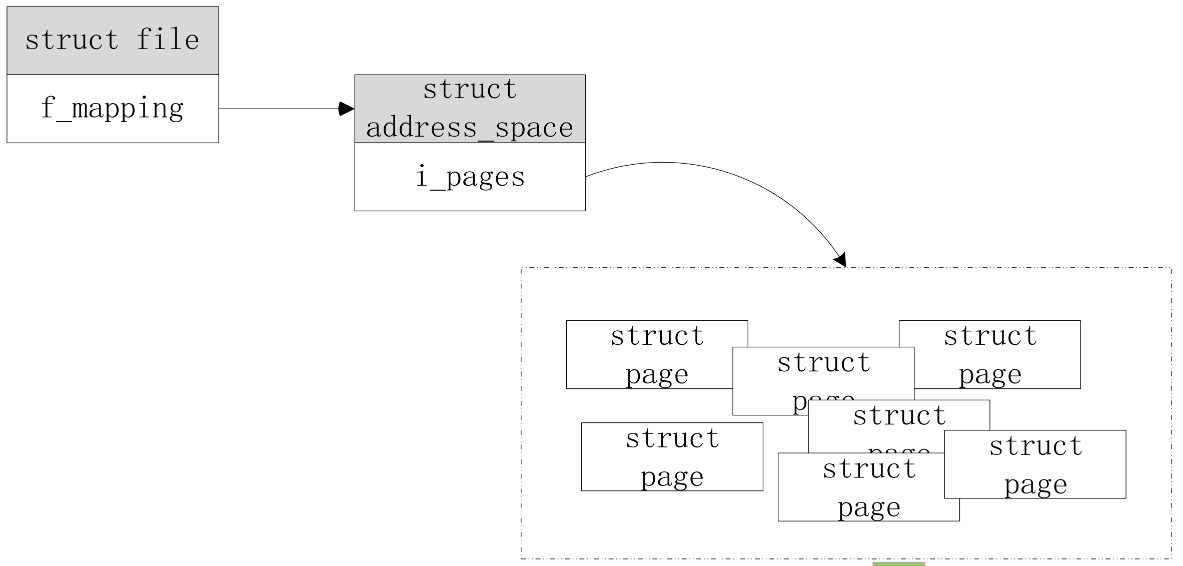
The flag releated to DAX mode



The changes of radix tree when enabling DAX mode

Non-DAX

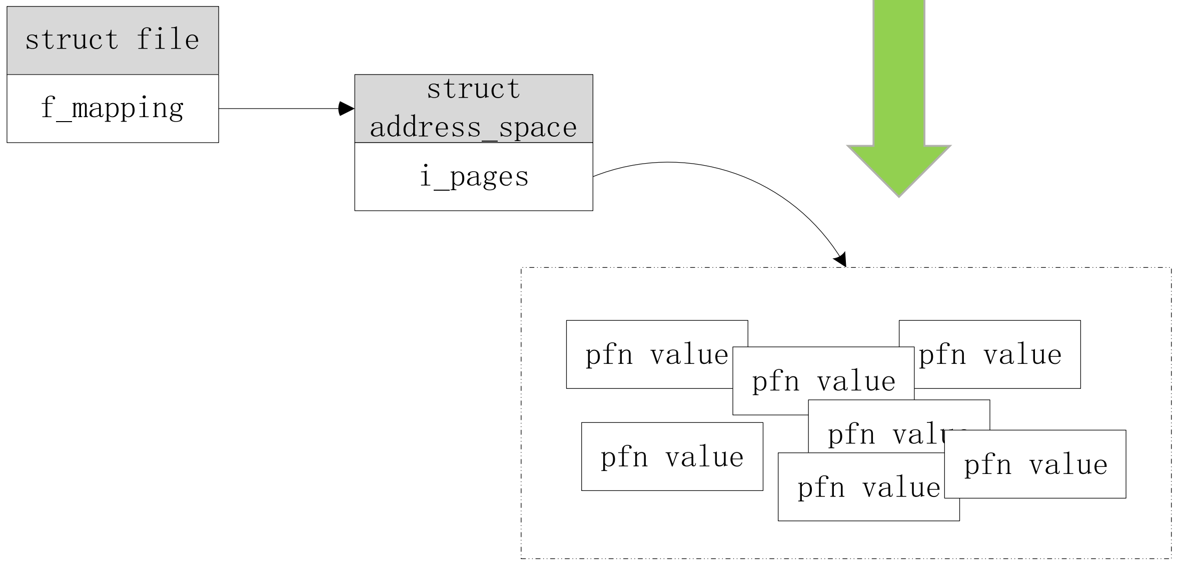
$ echo abcdefg > testfile



xfs_io –c ’chattr +x’ testfile

DAX

$ echo abcdefg > testfile

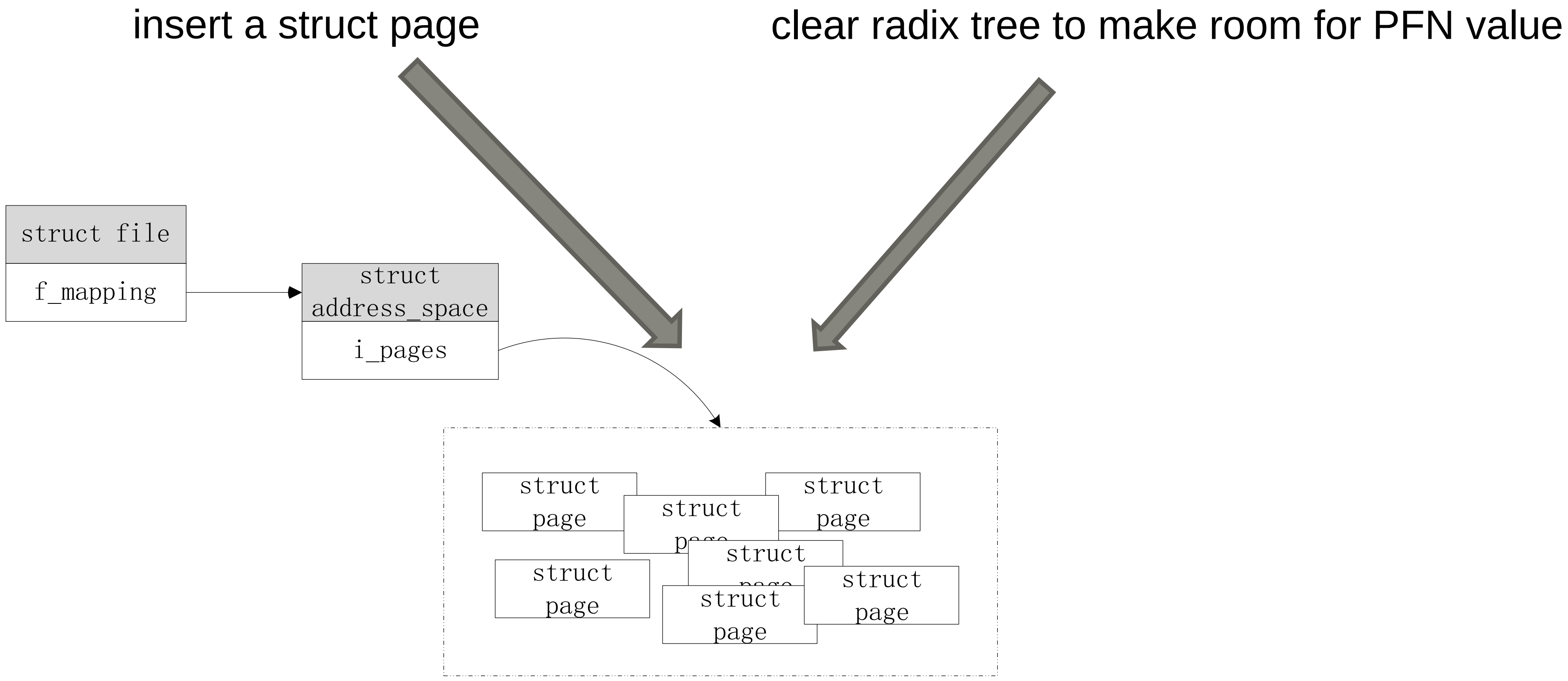


The race condition when switching radix tree

A thread is in page fault process and find DAX is disabled

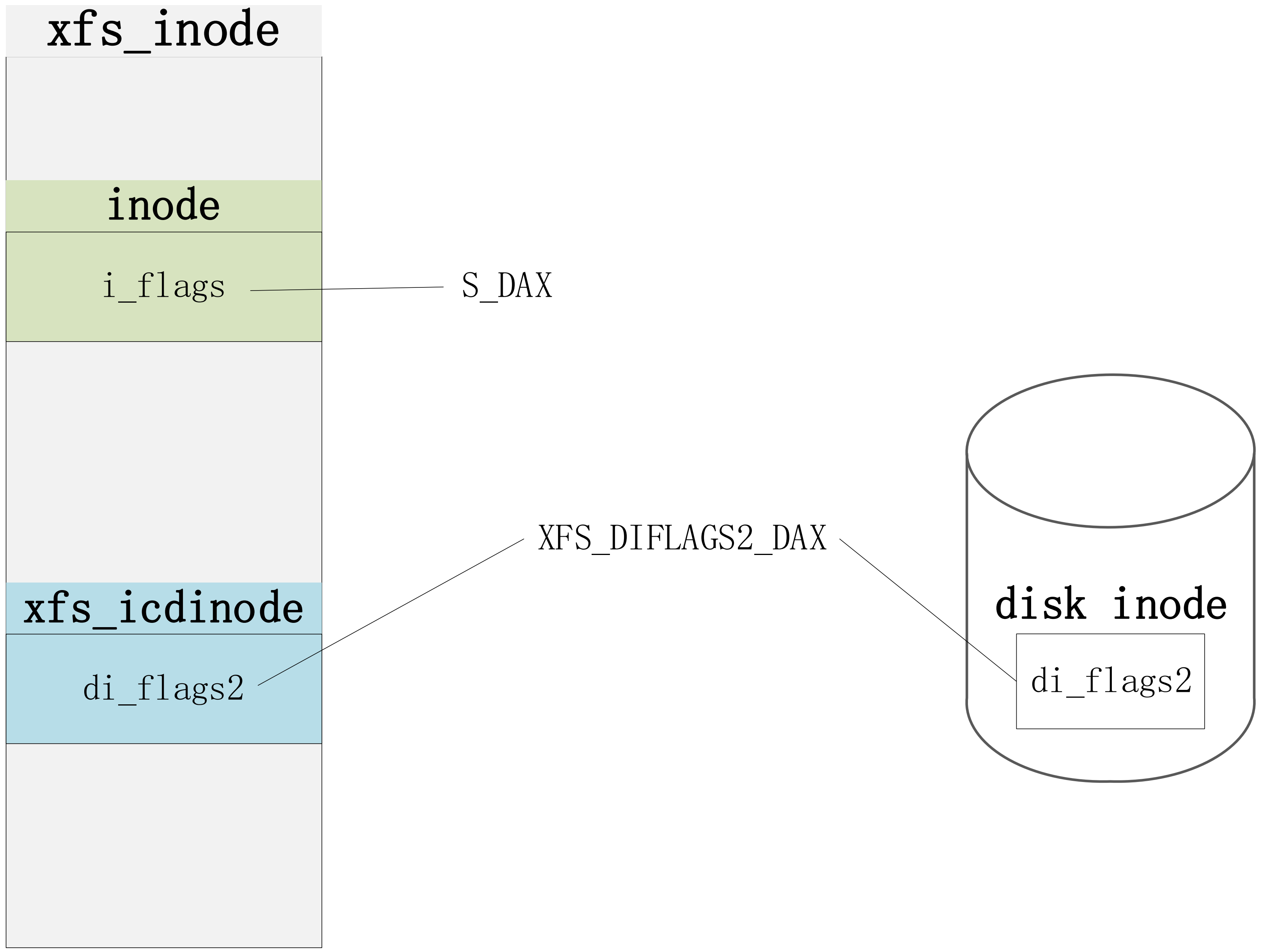
- alloc page frame in DRAM
- read file content from PMEM to DRAM page frame
- insert page struct to page cache radix tree

B thread use chattr +x to enable DAX

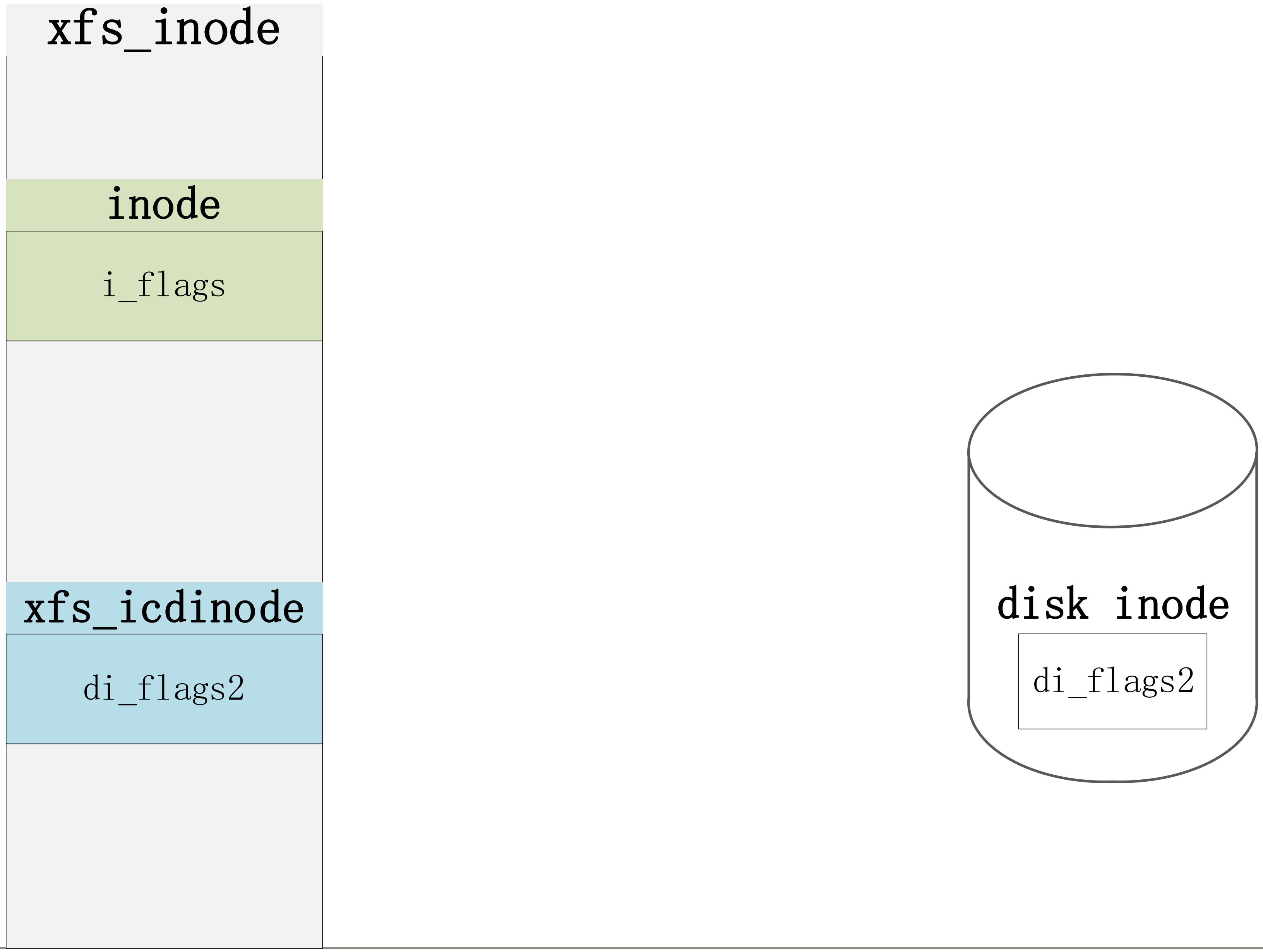


More details: http://lkml.iu.edu/hypermail/linux/kernel/1910.3/01067.html

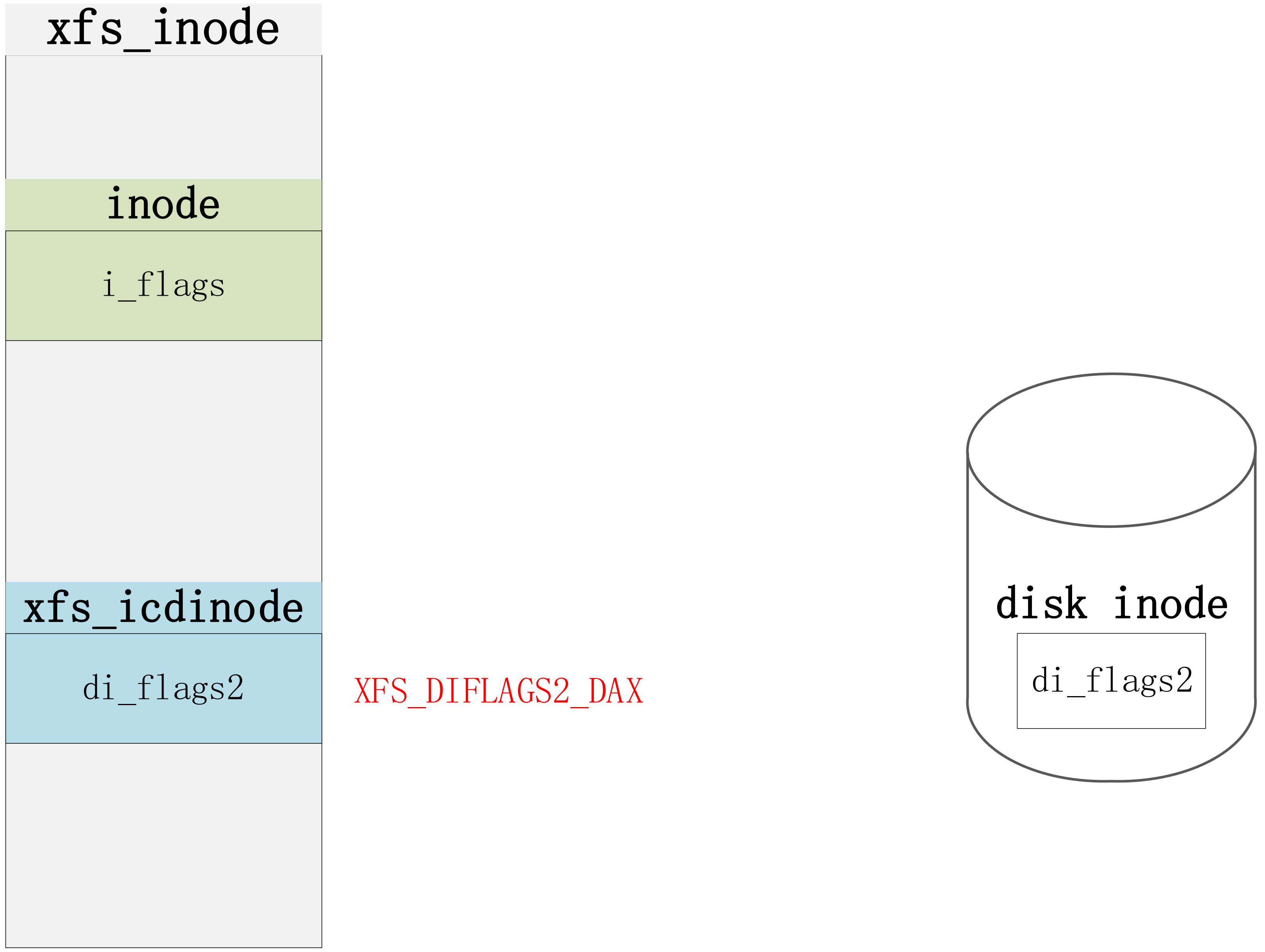
Two DAX-related flags



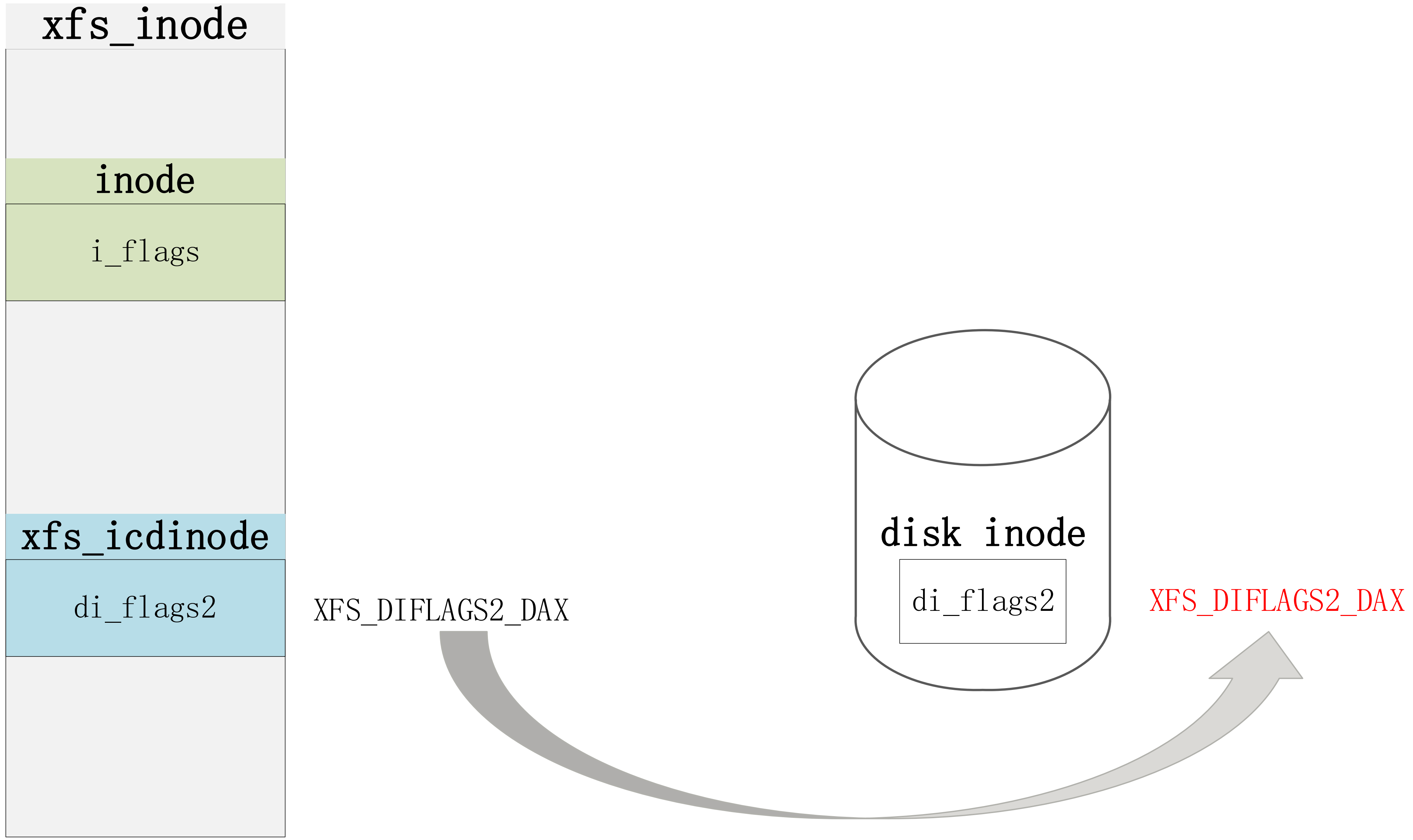
Initial state: Non-DAX



Enable DAX mode

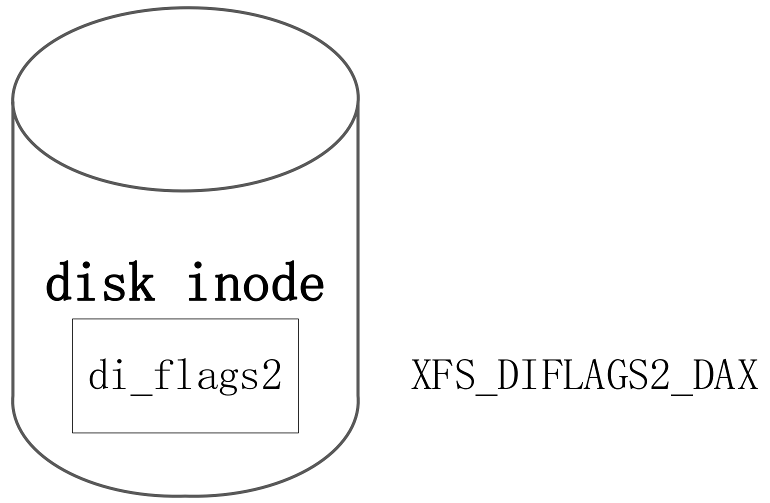


Sync xfs_inode to disk

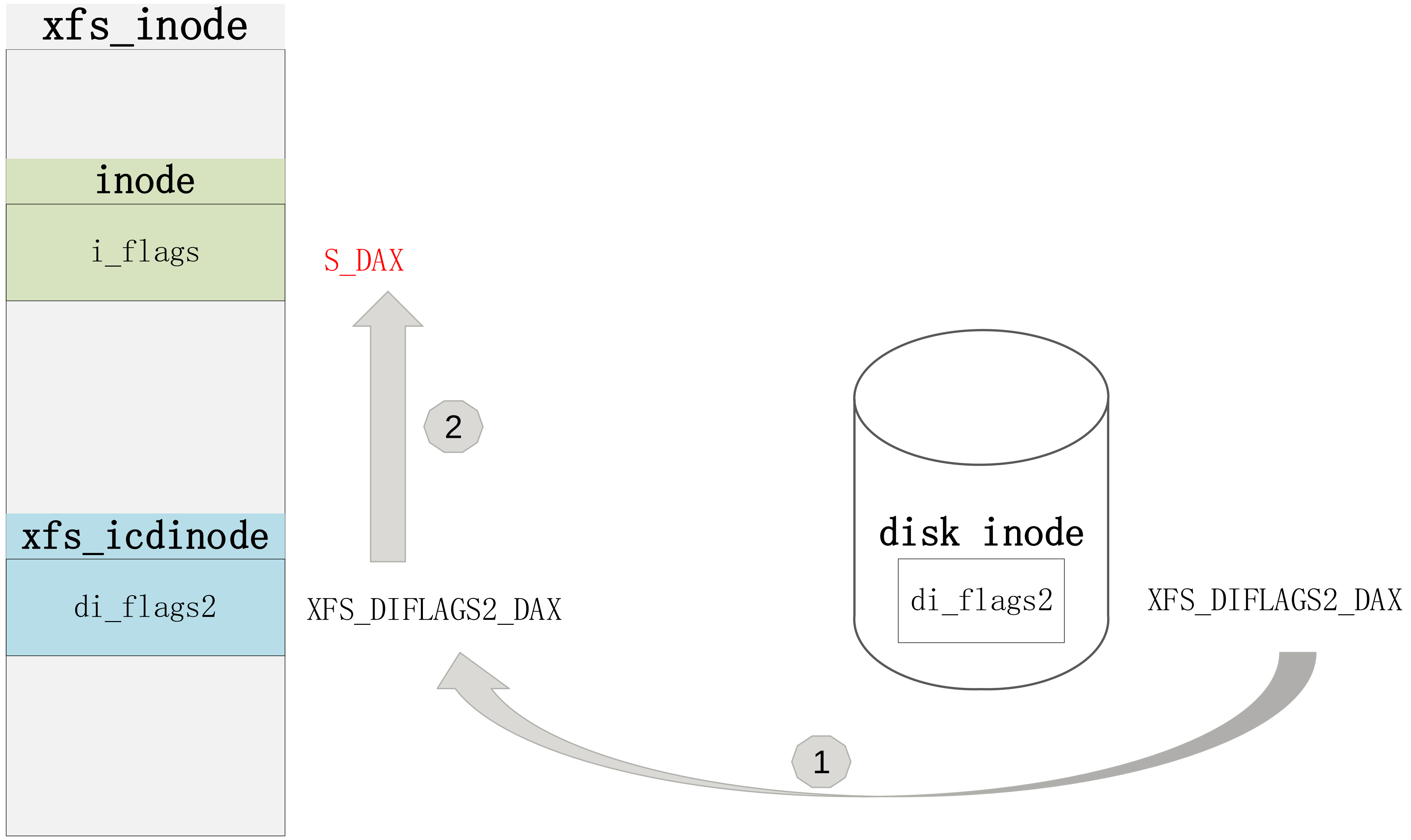


Evict inode from memory

Note: all process using this file must be terminated or they should close this file.



Re-Read inode from disk to memory



The details of drop_caches approach

- How to drop a specific inode from memory?
  - echo 2 > /proc/sys/vm/drop_caches
- Shortcomings
  - performance
  - inconvenience
    - permission

Two exist flags releated to free dentry/inode

- DCACHE_DONTCACHE
  - free dentry as soon as possible
- I_DONTCACHE
  - free inode as soon as possible

Problem1 statement

① *Non-DAX*

```
$ echo abcdefg > testfile
```

- **Close file**
  - dentry is inserted into LRU
  - set DCACHE_LRU_LIST on dentry

---

② *DAX*

```
$ echo abcdefg > testfile
```

- **Close file**
  - set DCACHE_REFERENCED on dentry

---

③ *Now, enable DAX mode*

```
$ xfs_io -c 'chattr +x' testfile
```

- **Enable DAX**
  - Set XFS_DIFLAG2_DAX
  - Set DCACHE_DONTCACHE on dentry
  - Set I_DONTCACHE on inode

- **Close file**
  - DCACHE_REFERENCED prevent dentry from being freed even though DCACHE_DONTCACHE is set

Solution for problem1

- If DCACHE_DONTCACHE is set, kill dentry unconditionally
  - https://lkml.org/lkml/2020/9/4/159

Problem2 statement

- If I_DONTCACHE is set, kernel will evict the inode without syncing the inode.
    - i_pages radix tree may have many dirty pages

Solution for problem2

- If I_DONTCACHE is set, sync inode before evicting it.
  - https://lkml.org/lkml/2020/9/24/56

Current approach

FUJITSU

1 *Non-DAX*

```
$ echo abcdefg > testfile
```

- **Close file**
  - dentry is inserted into LRU
  - set DCACHE_LRU_LIST on dentry

---

2 *DAX*

```
$ echo abcdefg > testfile
```

- **Close file**
  - set DCACHE_REFERENCED on dentry

---

3 *Now, enable DAX mode*

```
$ xfs_io -c 'chattr +x' testfile
```

- **Enable DAX**
  - set XFS_DIFLAG2_DAX
  - set DCACHE_DONTCACHE on dentry
  - set I_DONTCACHE on inode
- **Close file**
  - if DCACHE_DONTCACHE is set, kill dentry unconditionally
  - if I_DONTCACHE is set, sync inode and evict inode

---

4 *Open this file again*

- Open file
  - read disk inode to memory
  - S_DAX is set in inode because disk inode has XFS_DIFLAG2_DAX
  - Now we can say DAX is enabled for this file

FUJITSU

shaping tomorrow with you