Week 11.b
CS6640
11/16 2023
https://naizhengtan.github.io/23fall/

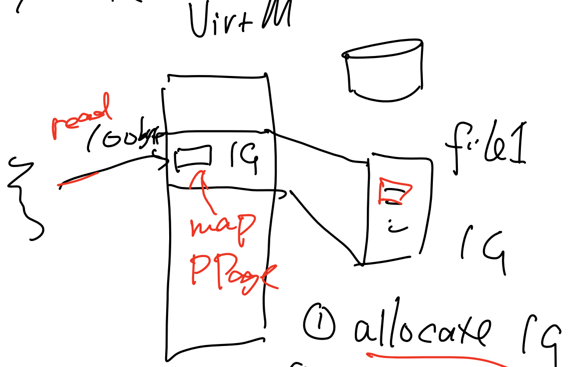
1. intro to fs
2. Unix files
3. what about now?
   -- PMem introduction
   -- modern(?) file mappings
---

last time: page fault

① open/read/write

② memory mapped files

↳ mmap



① allocate 1G

② Copy

mark 1G

remember the mapping

1. intr to FS. (basic)

Q: What does FS do?

- Save a document (WPS) modify it later
- reading from file (python)
- save data on disk (code)

HW

↳ ① persistence

file → ② name a seq of bytes (file)

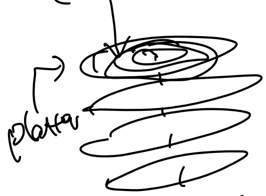
↳ ③ human-friendly-names (dirs)

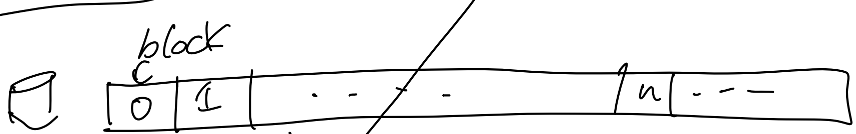
fs name space

Fast ← → slow

Classic: SSD / disk / tape

(track, sector)





array of blocks — 512B, 4KB/16KB

now: PMem

glass

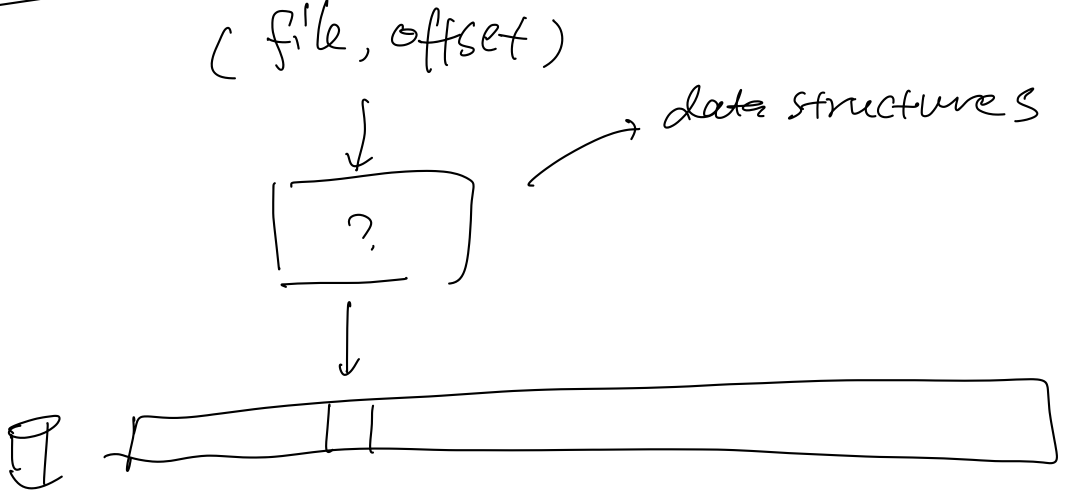
Silica

- CPU : scheduler ⟵ share CPU
- memory : Virtual memory ⟵ share memory
- disk : fs. ⟵ share disk

2. Files.

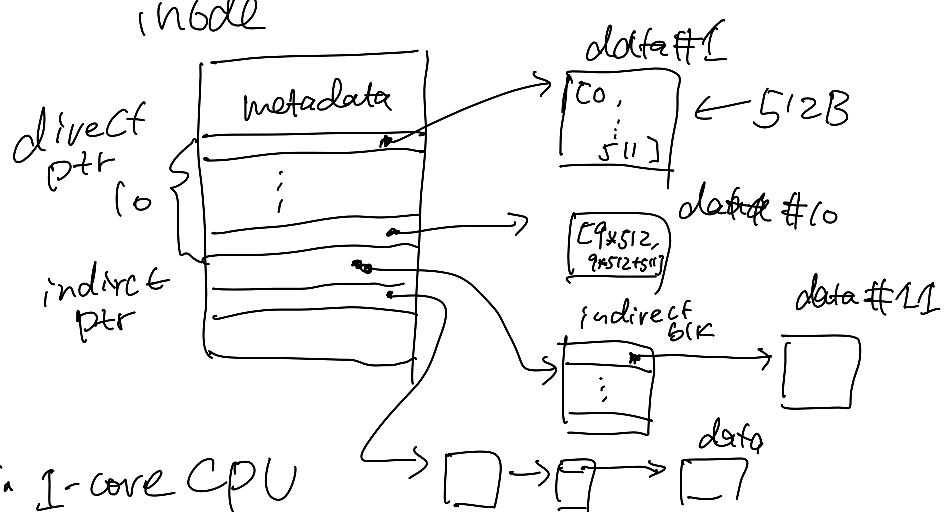
Q : What is a file
- user : bytes
- OS : a set blocks
- fs.

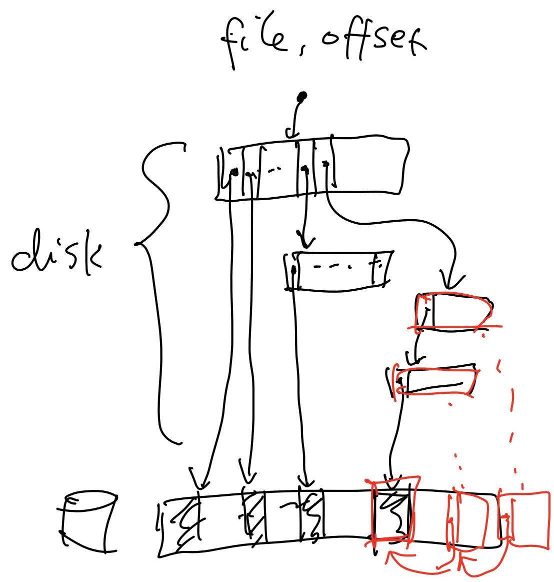
<ins>file mapping:</ins>



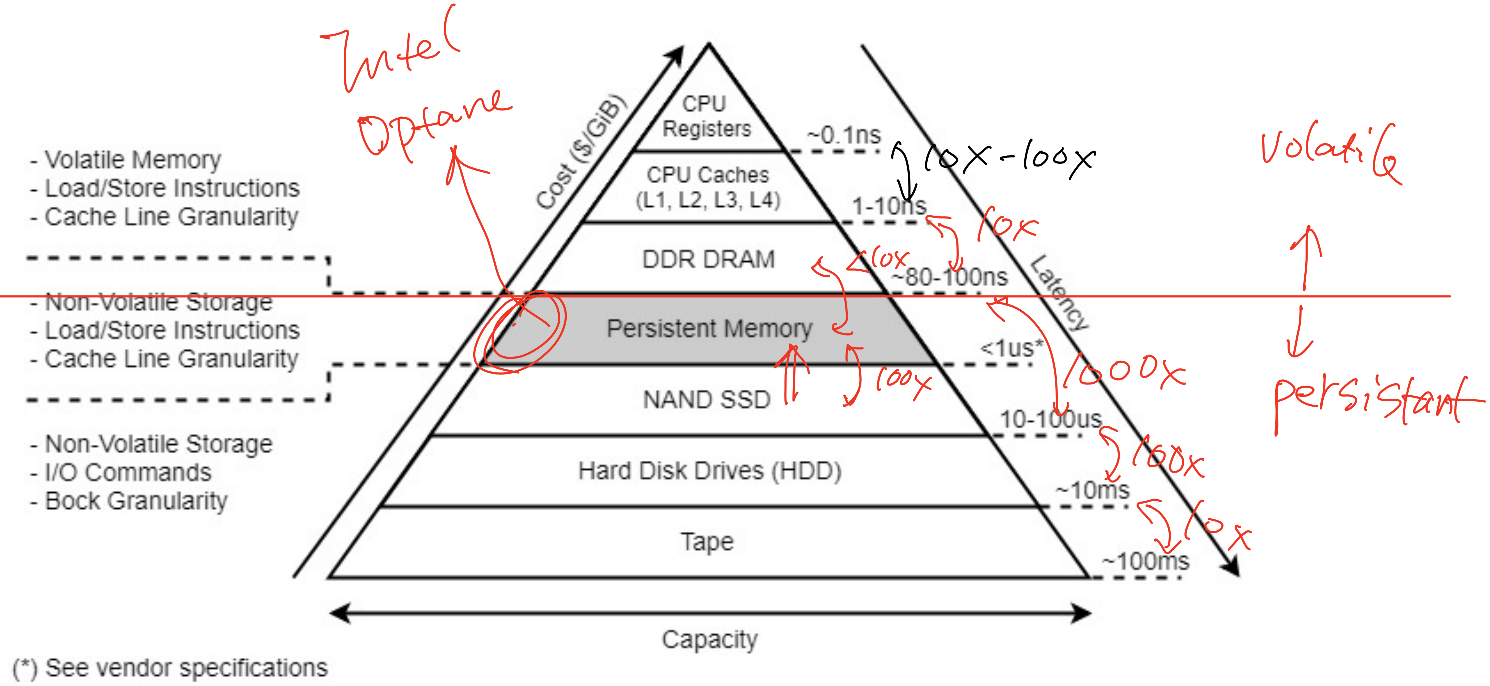
Unix file :

inode





- 1-core CPU
- IO slow (disk)
- seq access ⟹ random access



Figures borrowed from "PMDK Introduction"
https://docs.pmem.io/persistent-memory/getting-started-guide/what-is-pmdk

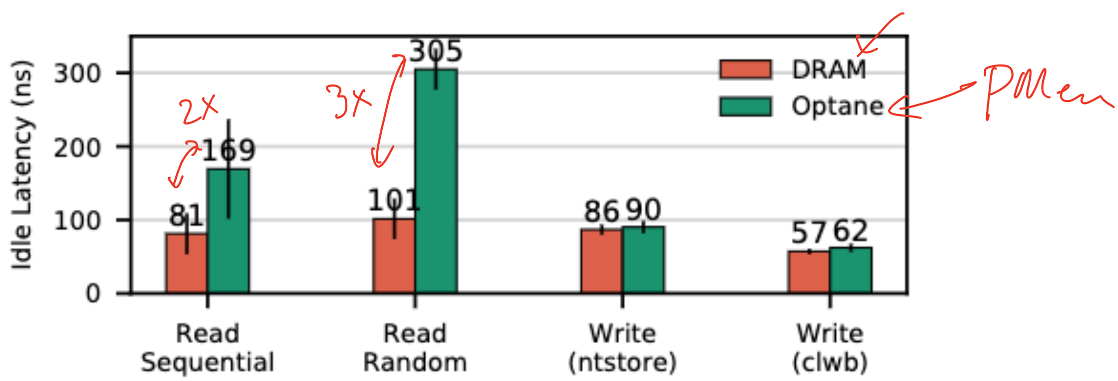


Figure 2: **Best-case latency** An experiment showing random and sequential read latency, as well as write latency using cached write with `clwb` and `ntstore` instructions. Error bars show one standard deviation.

Above figures are borrowed from An Empirical Guide to the Behavior and Use of Scalable Persistent Memory

Q: if your memory → persistent, do you need fs?

- Need. b/s names.

* FS design parameters:

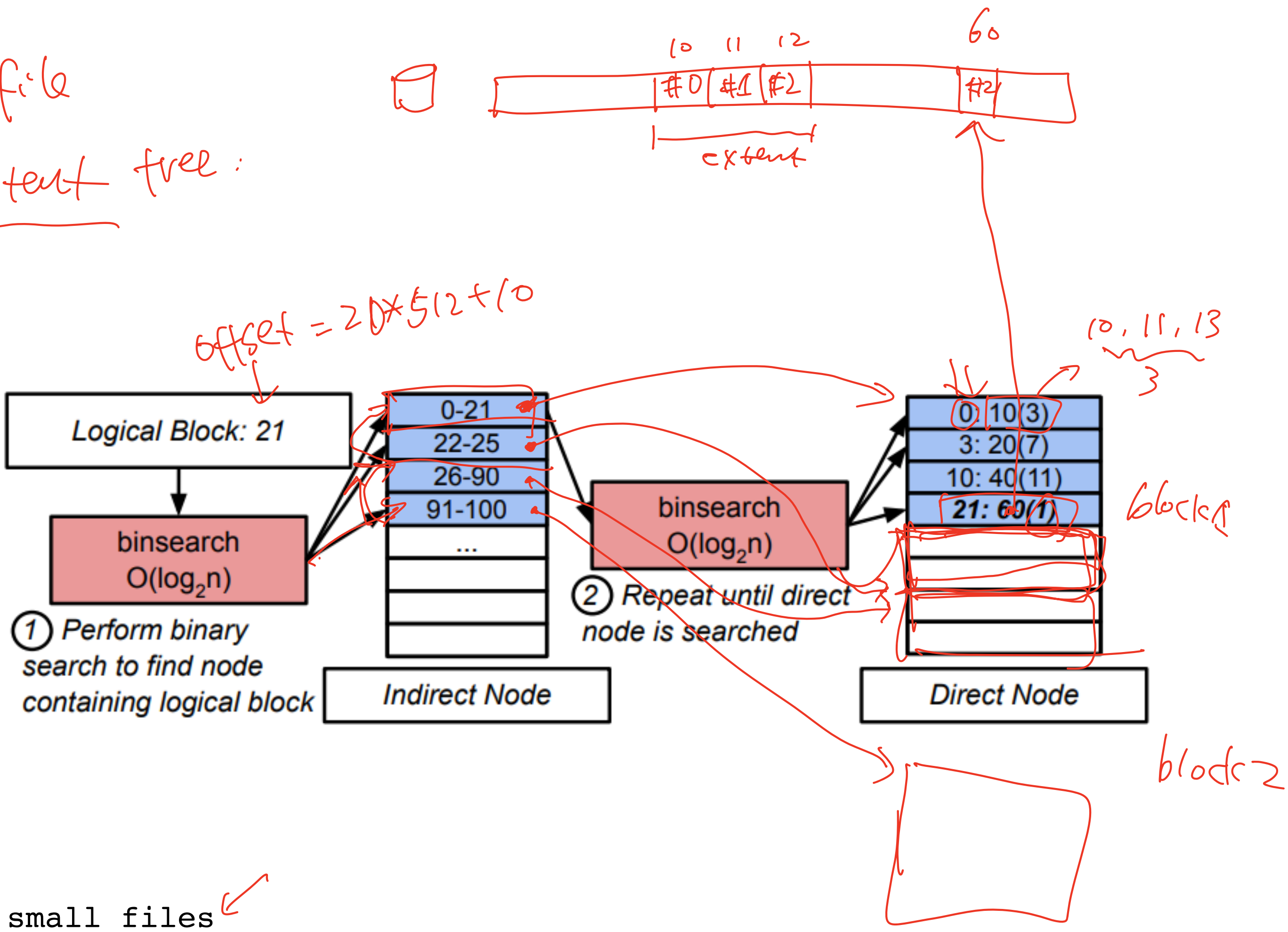
  - small files (most files are small)
      vs.
    large files (much of the disk is allocated to large files)

  - sequential access vs. random accesses

  - prefetching

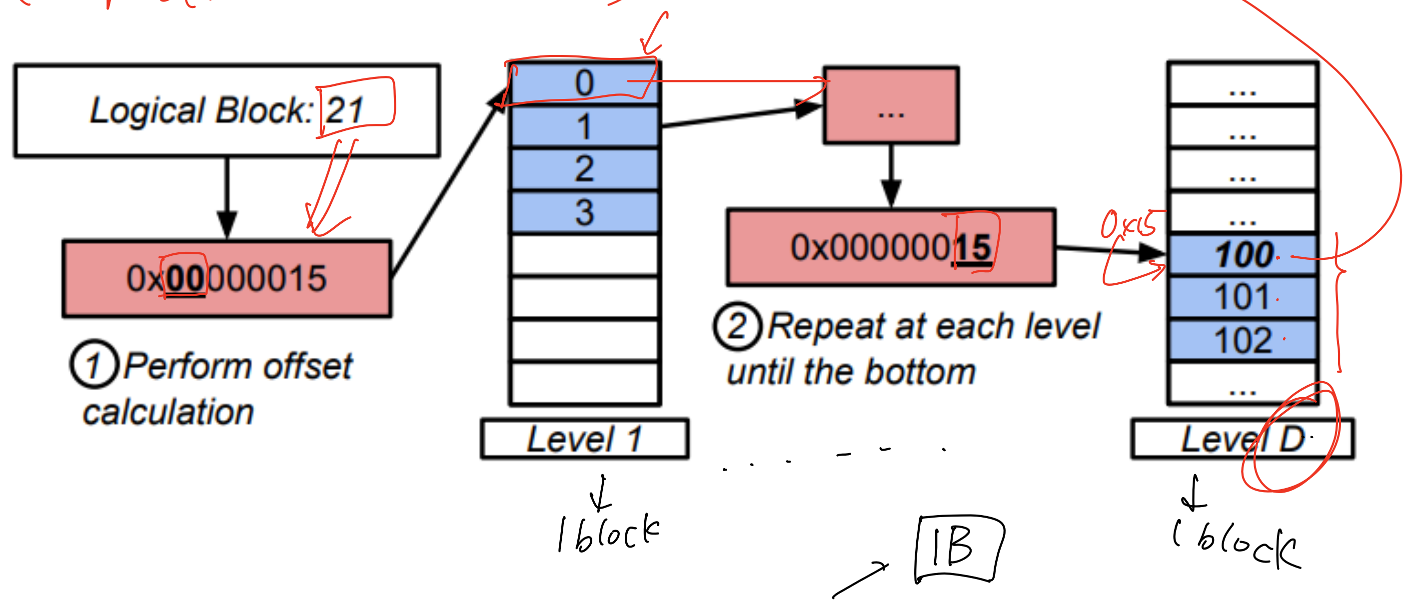
  - disk utilization (metadata overhead and fragmentation)

per-file extent tree:



    [ ] small files
    [ ] large files
    [ ] sequential access
    [ ] random accesses
    [ ] prefetching
    [ ] disk utilization

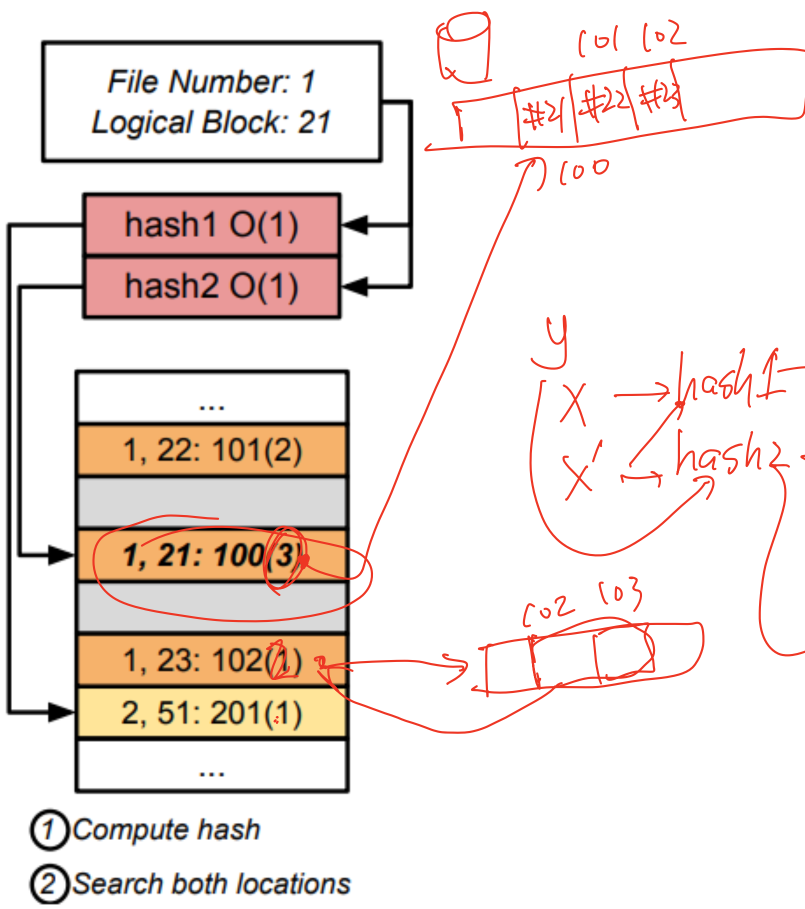
per-file radix tree (PT)



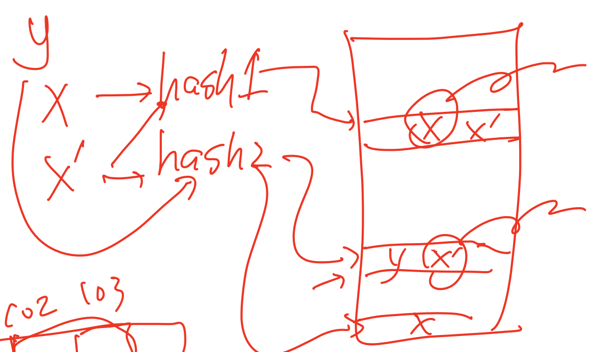
D blocks

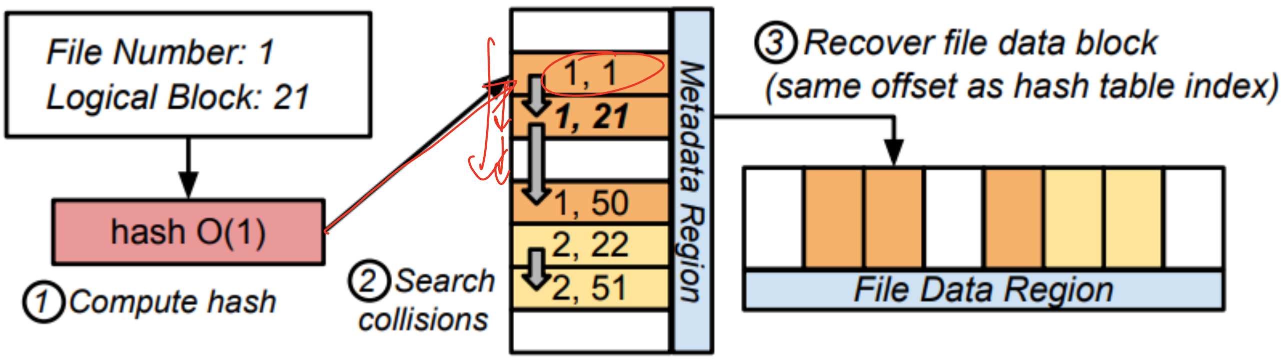
- [X] small files
- [ ] large files
- [ ] sequential access
- [ ] random accesses
- [ ] prefetching
- [ ] disk utilization

Global Cuckoo hash table



- [✓] small files
- [✓] large files
- [ ] sequential access
- [✓] random accesses
- [ ] prefetching
- [ ] disk utilization






```
[   ] small files
[   ] large files
[   ] sequential access
[   ] random accesses
[   ] prefetching
[   ] disk utilization
```