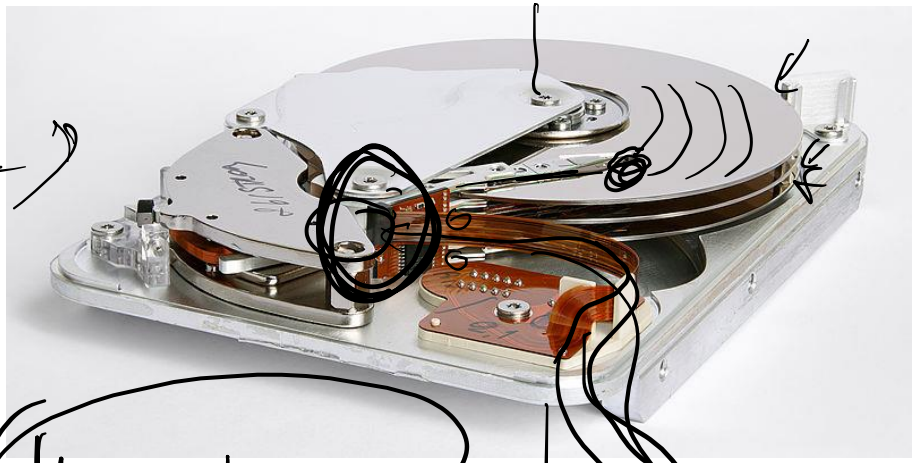
Fast File Systems

How to improve the performance & reliability of the basic file system?

ext2
ext3

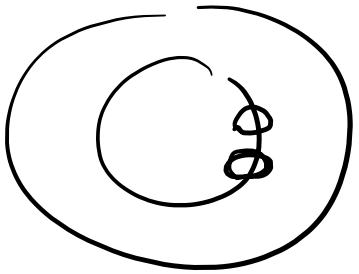
What is the problem?

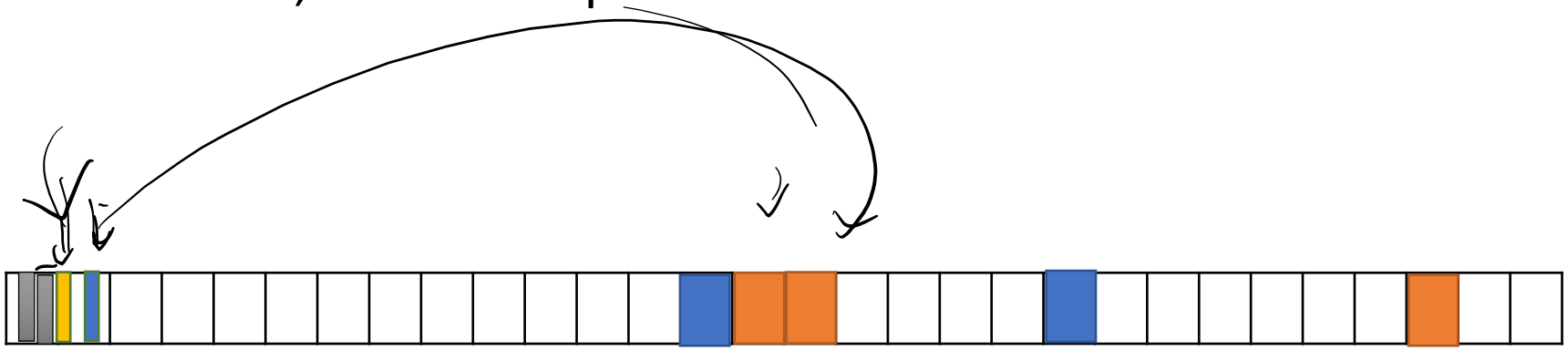
- Reliability
- Performance
  - How to read/write a file a/b?



What is the problem?

- Reliability
  - Most important data all on one surface
- Performance
  - Too much seek and rotational delays!
  - Too much random accesses; too few sequential accesses!





Solution for improving performance

- Change block sizes
  - 512 B → 4K

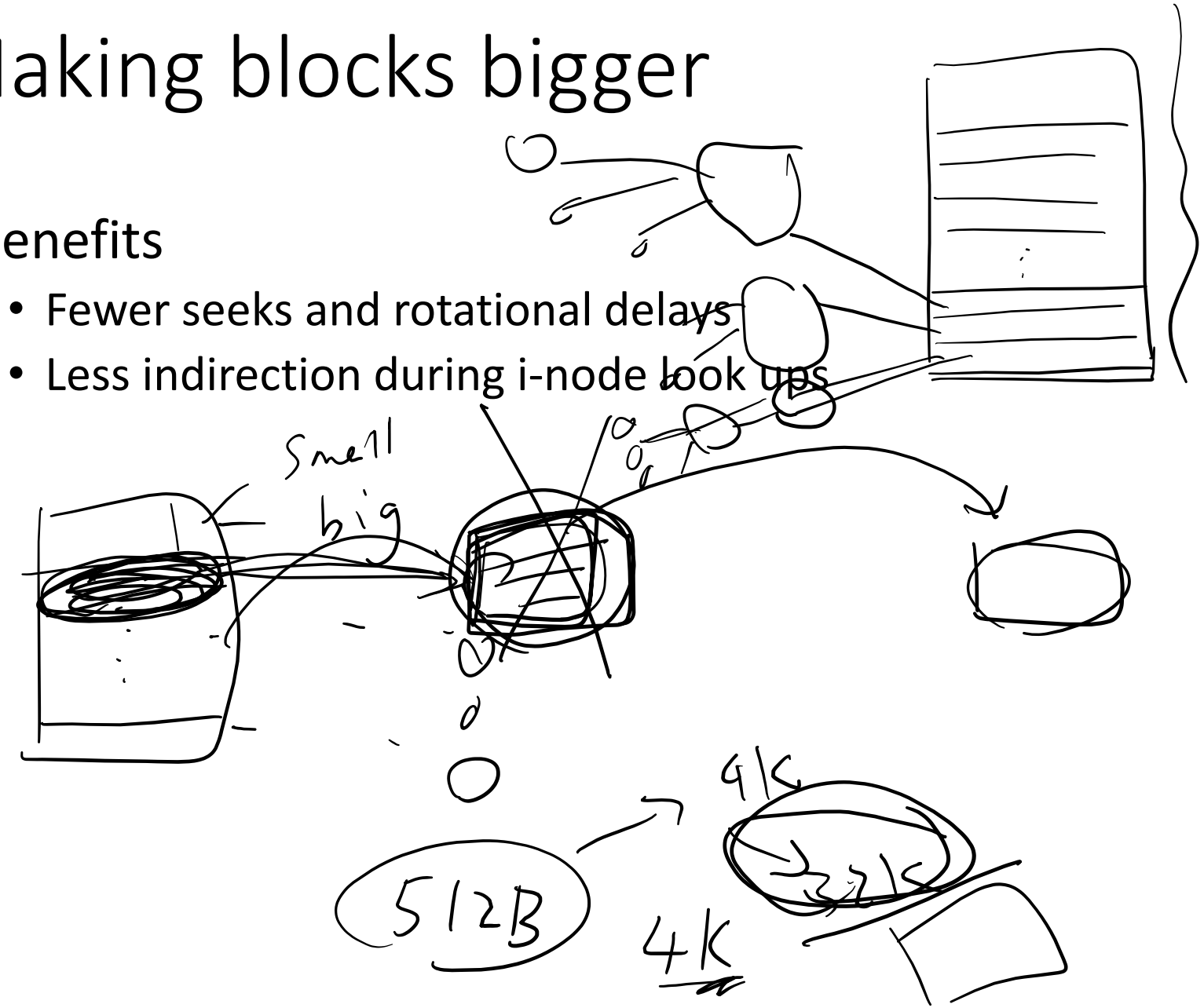
- Change data layout

Question

- What type of block sizes tend to incur fewer seeks?

Making blocks bigger

- Benefits
  - Fewer seeks and rotational delays
  - Less indirection during i-node look ups



Making blocks bigger

- Benefits
  - Fewer seeks and rotational delays
  - Less indirection during i-node look ups
- Limitations
  - Waste disk spaces
  - Potentially hurt the effectiveness of buffer cache

Solutions in FFS

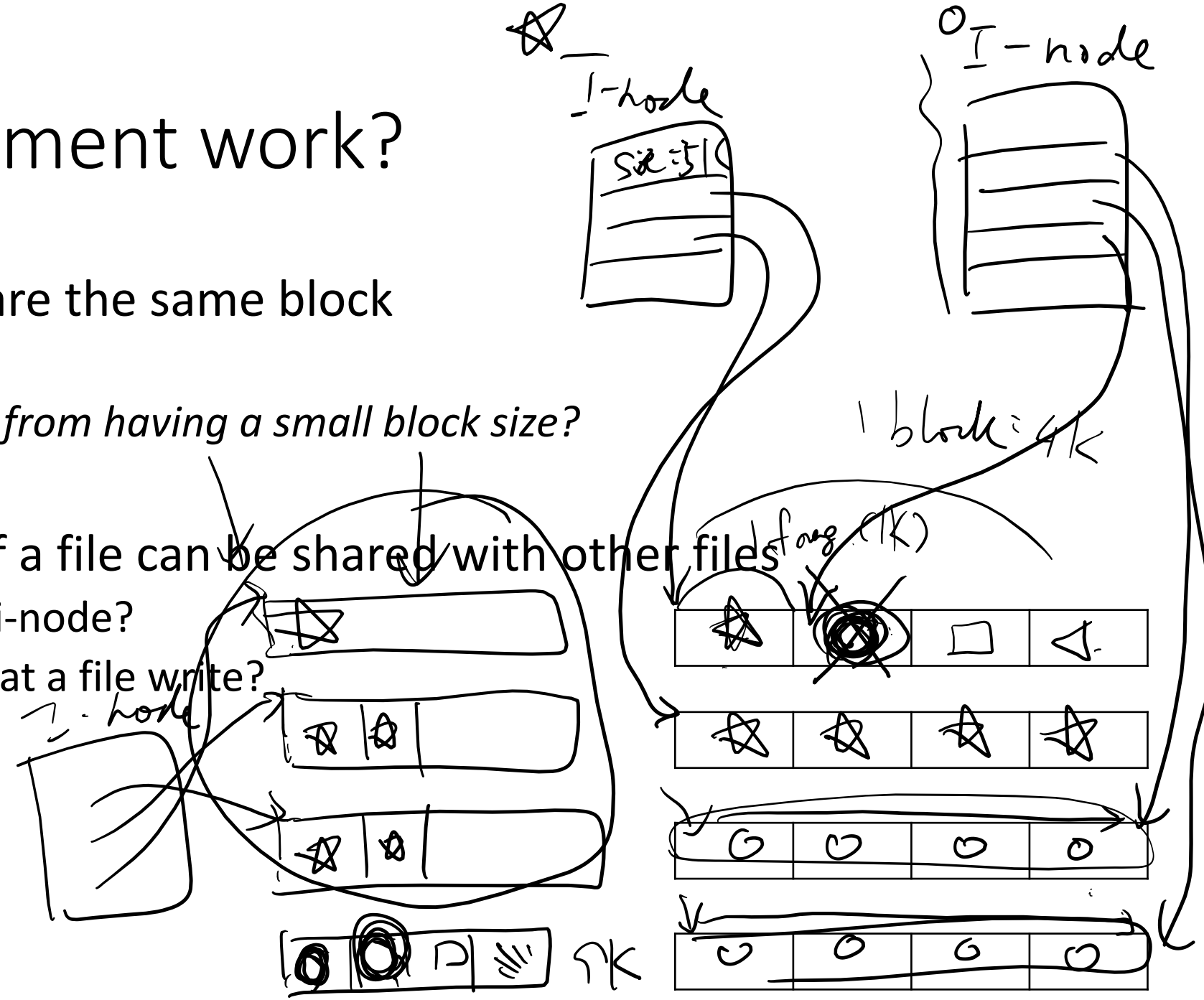
- Block size increases from 512B to 4K
- Use “fragment” (512B) as disk allocation unit

How does fragment work?

- Multiple files can share the same block

  What is the difference from having a small block size?

- Only the last block of a file can be shared with other files
  - What are inside the i-node?
  - What might happen at a file write?



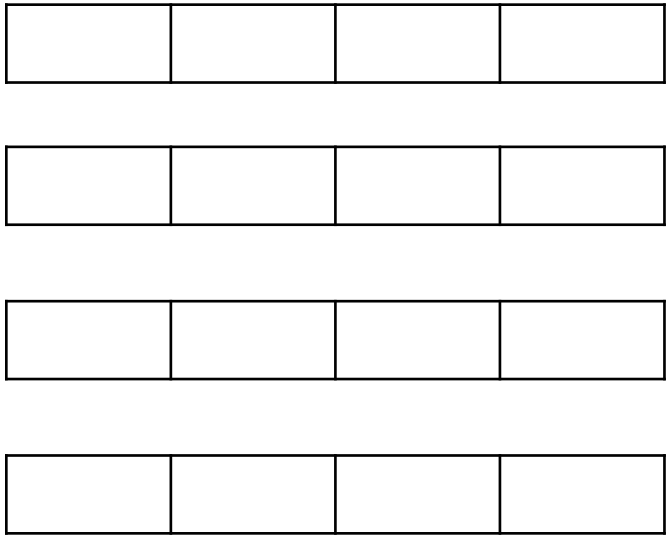
How does fragment work?

- Multiple files can share the same block

  What is the difference from having a small block size?

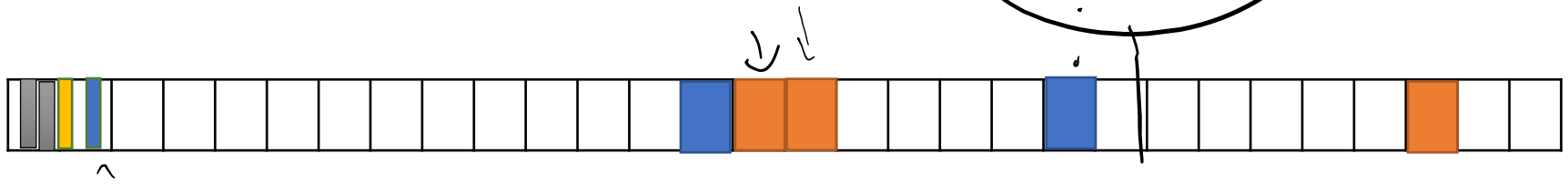
- Only the last block of a file can be shared with other files
  - What are inside the i-node?
  - What might happen at a file write?

  + space saving
  + fewer seeks and rotational delays
  - Write can be very slow



How to improve the data layout?

- How to promote sequential accesses?
  - Put blocks that are likely to be accessed together (temporal) close to each other (spatial) on the disk



Question

- What blocks tend to be accessed together (temporally)?

What blocks need locality?

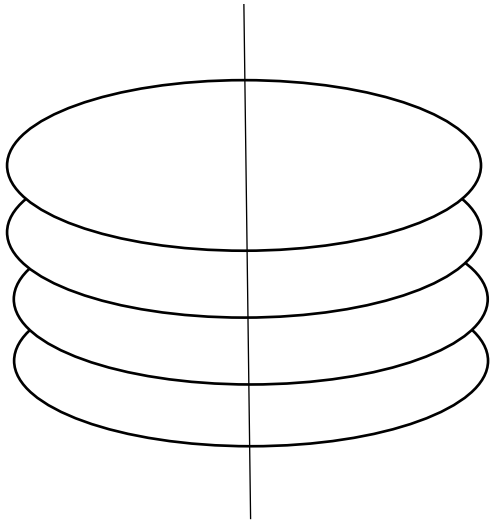
- Data blocks of the same file
- A file’s data block and the file’s i-node
- Data blocks of files under the same directory

How to support locality?

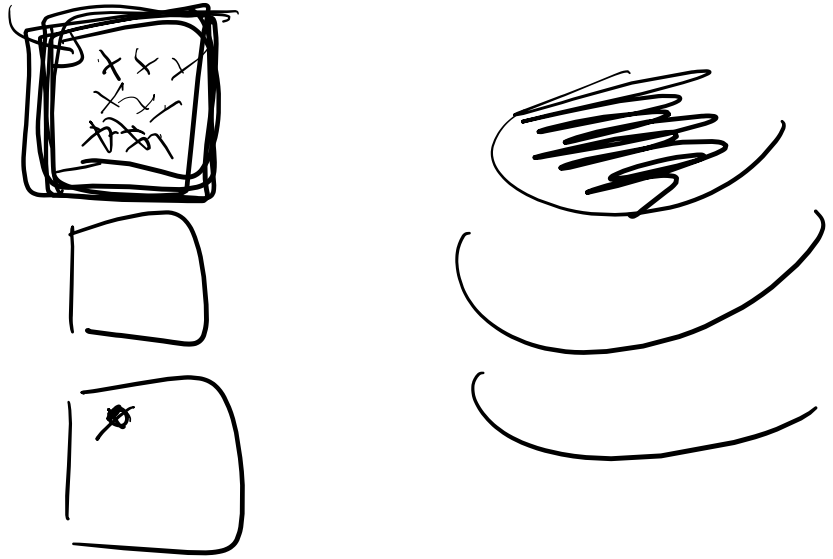
- Data blocks of the same file
- A file's data block and the file's i-node
- Data blocks of files under the same directory

- Can we put a file's data block strictly sequential?
- Can we put a file's i-node together with its data blocks?

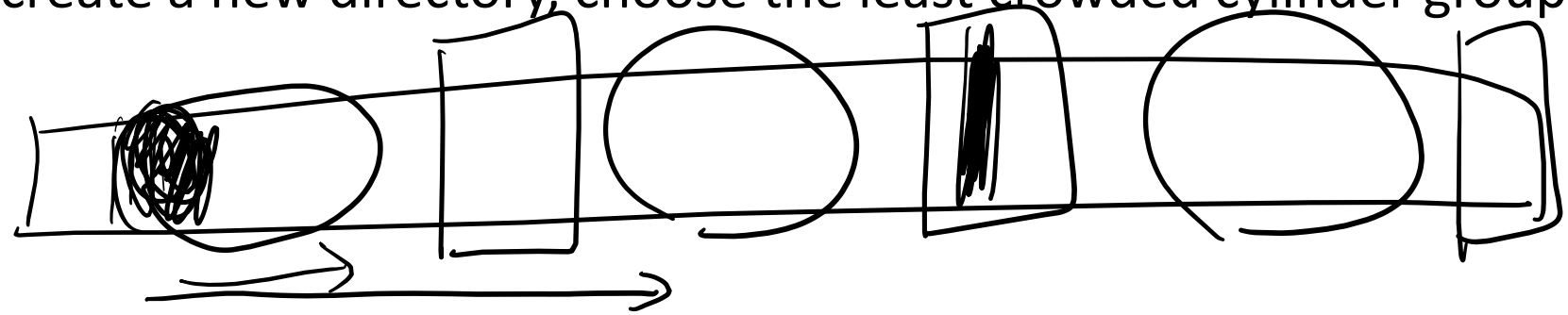
Cylinder group



Layout with cylinder groups



- Data blocks of the same file
  - ➔ Same cylinder group (unless they spill)
- A file's data block and the file's i-node
  - ➔ Same cylinder group (i-nodes all at the beginning of a cylinder group)
- Data blocks of files under the same directory
  - ➔Same cylinder group (unless they spill)
  - ➔When we create a new directory, choose the least crowded cylinder group



What about reliability?

- How to make sure important meta data is not on the same surface?
  - Replicate super blocks
  - Put i-nodes at different locations in each cylinder groups