5/17/24 More Journaling

all updates reflected or no updates

→ Transaction : group arbitrary # of updates into an atomic unit

→ write txn in the log : txn_begin | changed blocks ... | tx_commit

☆ apply changes once the committed txn is persisted on disk

☆ upon reboot, recover by replaying all committed txns in the log

☆ Data Journaling → txn records both data & metadata costly : once to the log, once to actual location

→ strong crash safe guarantees, atomic updates of data, consistent fs metadata

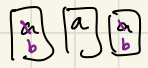
☆ Metadata Journaling → txn only log changes to metadata (bitmaps, inodes, directory entries)

→ upon fsync, writes data directly to their actual location

→ then persist the log w/ metadata txn

→ default ext4 journaling mode (ordered)

relaxed data guarantees, may see a mixture of old & new data content



a = old data

b = new data

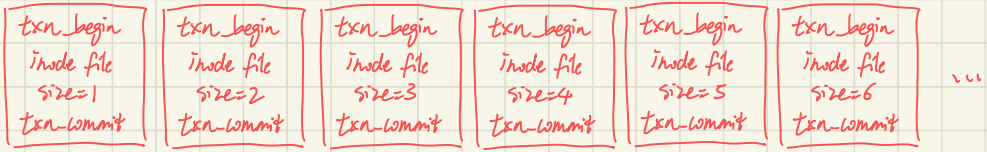
→ Transaction Granularity

→ 1 fs op ⇒ 1 txn

→ only ops that modify the file system need to be logged.

→ conceptually intuitive but lots of unnecessary writes

→ e.g. write(fd, "a", 1) x 100



inefficient: records intermediate state of shared metadata

→ compound txn: group multiple fs ops into one txn

→ one global active txn: tracks a list of changed metadata (ptr to cached inodes, bitmap blocks)

closed/committed periodically (30s)

or on fsync ⇒ while committing the current txn, start a new global one for incoming fs requests

need to first persist the data blocks in the txn & then persist the txn.

★ fsync performance affected by unrelated files in the same txn:

e.g. mail server (small writes + fsyncs) with database (lots of writes)

→ Physical vs Logical Logging

↳ log higher level operation (add extent x), derive changes to data bitmap

longer/more complex recovery process, much easier to disentangle changes from different operations.

↓

logs updated physical content, easy to replay, hard to disentangle effects from different operations

→ Fast Commits in ext4

→ certain ops support logical logging

→ on fsync, only persist the requested file's data & persist the logical ops. for txn

☆ no interference from other files, faster overall!