

Homework 12 — December 19

Lecturer: Jiang Dejun

Completed by: 2022K8009929010 Zhang Jiawei

13.1

假设一个块的大小为 4KB, 则创建 /home/OS24/fs03.pdf 时, 需要修改位图块、创建 fs03.pdf 的 inode 块、修改目录块、修改 /home/OS24 的 inode 块、添加 3 个数据块并写入、再次修改 fs03.pdf 的 inode 块, 共需要修改 7 个块。

下面列举出任意时刻宕机后出现不一致的情况:

- 在修改位图块后宕机, 此时位图块已经被修改, 但是其他块没有被修改, 会导致位图块中的数据块被分配但是其他块没有被修改。
- 在创建 fs03.pdf 的 inode 块后宕机, 此时 fs03.pdf 的 inode 块已经被创建, 但是并没有修改目录块, 会导致 fs03.pdf 的 inode 块被创建但是目录块中并没有 fs03.pdf 的记录。
- 在修改目录块后宕机, 此时目录块已经被修改, 但是并没有修改 /home/OS24 的 inode 块, 会导致目录块中的 fs03.pdf 记录被修改但是 /home/OS24 的 inode 块中并没有 fs03.pdf 的记录。
- 在修改 /home/OS24 的 inode 块后宕机, 此时 /home/OS24 的 inode 块已经被修改, 但是数据块并没有被修改, 导致元数据与数据不一致。
- 在添加数据块并写入后宕机, 此时数据块已经被写入, 但是 fs03.pdf 的 inode 块并没有被修改, 导致数据块中的数据没有被记录。

13.2

- (1) 若在日志完成 TxE 之前宕机, 则磁盘内日志不完整, 什么也不做, 文件 A 的内容仍为旧的内容; 若在日志完成 TxE 之后宕机, 由于磁盘内日志完整, 则按照日志中的内容进行恢复, 文件 A 的内容为新的内容。
- (2) 若在日志写 TxB 前宕机, 则元数据不变, 新写入的数据视为垃圾块, 文件 A 的内容仍为旧的内容; 若在日志写 TxB 后写 TxE 前宕机, 元数据日志不完整, 则元数据不变, 新写入的数据视为垃圾块, 文件 A 的内容仍为旧的内容; 若在日志写 TxE 后宕机, 由于日志完整, 按照日志中的内容进行恢复, 元数据更新, 文件 A 的内容为新的内容。

13.3

- (1) 每个 imap 可以索引到 $\frac{4000}{4} = 1024$ 个 inode, 因此需要 $\lceil \frac{2000000}{1024} \rceil = 1954$ 个 imap 块。
- (2) 每个 CR 可以索引到 $\frac{4000}{4} = 1024$ 个 imap 块, 因此需要 $\lceil \frac{1954}{1024} \rceil = 2$ 个 CR 块。
- (3) ino=356302 的 inode 在第 $\lceil \frac{356302}{1024} \rceil = 348$ 个 imap 块的第 $356302 - 1024 \times 347 = 974$ 个位置。因此只需读取第一个 CR 块的 CR[347][973] 即可找到 ino=356302 的 inode。

13.4

- (1) 直接指针的索引范围为 4KB, 一级间接指针的索引范围为 $4 \times \frac{4 \times 1024}{4} = 4MB$, 二级间接指针的索引范围为 4GB, 三级间接指针的索引范围为 4TB。故文件 foo 使用了直接指针、一级间接指针、二级间接指针。
- (2) 文件 foo 总共占用的磁盘块数为 $\frac{10 \times 1024}{4} = 2560$ 块, 则需要写入的为 foo 的最后一块。写流程为: 输入数据写数据块、写 inode、更新 imap, 共 3 次 I/O 操作。
- (3) 同样写最后一块, 但是采用 FFS 的写流程为: 写数据块、写 inode、更新 bitmap, 共 3 次 I/O 操作。
- (4) 如果是日志文件系统, 则增加写 TxB、写修改操作、写 TxE, 共 6 次 I/O 操作。