Homework Wan Huzaifah bin Wan Azhar

Answer:



* Commands of -n 3 -s 3 in order:
  + Create file /jp6
  + Create file /vg2
  + Link file /jp6 /mq1
* Live file:
  + 13, 14, 10 from final imap
  + 12 from 13
  + 15 is final imap
  + 0 is checkpoint
  + So, 0, 10, 12, 13, 14, 15
* Considerably harder as there is more indirection, and liveness of each file is difficult to determine. However, it is not impossible, just need a bit more patient.



* -n 5 -s 4 -i
* Commands in order:
  + Create file /ke0
  + Write file /ke0
  + Create file /he1
  + Creat file /vu8
  + Create file /qt8



* -o -F -s 100
* Commands:
  + create file /us7
  + write file /us7 offset=4 size=0
  + write file /us7 offset=7 size=7
* Final state:

INITIAL file system contents:

[ 0 ] live checkpoint: 13 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

[ 1 ] [.,0] [..,0] -- -- -- -- -- --

[ 2 ] type:dir size:1 refs:2 ptrs: 1 -- -- -- -- -- -- --

[ 3 ] chunk(imap): 2 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

[ 4 ] live [.,0] [..,0] [us7,1] -- -- -- -- --

[ 5 ] live type:dir size:2 refs:2 ptrs: 4 -- -- -- -- -- -- --

[ 6 ] type:reg size:0 refs:1 ptrs: -- -- -- -- -- -- --

[ 7 ] chunk(imap): 5 6 -- -- -- -- -- -- -- -- -- -- -- -- -- --

[ 8 ] type:reg size:0 refs:1 ptrs: -- -- -- -- -- -- --

[ 9 ] chunk(imap): 5 8 -- -- -- -- -- -- -- -- -- -- -- -- -- --

[ 10 ] live usususususususususususus

[ 11 ] live type:reg size:8 refs:1 ptrs: 10 -- -- -- -- -- --

[ 12 ] live chunk(imap): 5 11 -- -- -- -- -- -- -- -- -- -- -- -- -- --



* Live blocks:
  + 98 = 97
  + 80 = 34, 68, 69, 70, 76, 77, 78, 79
  + 66 = 60, 64, 65
  + 94 = 93
  + 91
  + 95
  + 99
  + 0
* Valid pathnames:
  + Live dir: /ln7
  + Live files: /lt0, /af4, /ln7/zp3, /ln7/zu5



* Parameter: -L c,/foo:w,/foo,0,1:w,/foo,1,1:w,/foo,2,1:w,/foo,3,1 -o
* Live blocks:
  + 0, 5, 18, 4, 8, 11, 14, 17, 19



* Parameters: -L c,/foo:w,/foo,0,4
* Live blocks:
  + 0, 4, 5, 8, 9, 10, 11, 12, 13
* Buffering improves sequential writes by merging all writes into one single write.
* As we can see from live blocks, the writes are done sequentially from 8 to 12, which improves reads as well.



* -L c,/foo:w,/foo,0,1 is different from -L c,/foo:w,/foo,7,1 because the write of /foo in the first case starts from offset 0 while the writes in second case starts from offset 7
* As such, second file has higher size because it starts from 7 (and thus previous blank lines has to be wrote) and writes data once.



* A new data block will be allocated if new directory is created. The inode created is also a directory inode.
* Aside from that, everything else is the same.



* No block is created, instead, the file system will just update the references to the inode.



* Sequential allocation of inode number will leads to sequential imap, means that less imap is required to allocate node.
* Random allocation of inode number will leads to many different imap pieces, because imap are grouped according to their types, as in a directory has one imap and file has its imap, etc.
* In short, every imap is random.



* If checkpoint is never updated, the initial checkpoint would only lead to root directory. Hence, everything else is considered as garbage.
* If checkpoint is updated periodically, then it will have the latest imap and by extension, address to the data.
* There is a problem: the textbook stated that checkpoint is updated every 30 seconds.
  + What if a data is wrote before 10 seconds and another program wants to access the data before 30 seconds?
  + Won’t that trigger an exception because the checkpoint is not pointing to the latest imap?
  + The system cannot roll-forward because this is not considered a crash and doing so would mess with performance. I,e, imagine if every 30s, a program try to access data that is already been written, the system will have to roll-forward every times and making periodic timer redundant.
  + Maybe the solution to this problem is to read imap in the temporary memory first before persistent disk?
* Roll forwarding in -N -I -o -s 1000 would means retrying step 4 to 17.