

CSE Recitation-2 20161003

Name: ChenJiayang ID: 5140379036

Question1: What is the problem this paper tried to solve? (Hint: please state the problem using data)

This paper is trying to reimplement a new file system which is faster than the old one, including these two aspects:

- **Low throughput.** Because now the original UNIX file system is incapable of providing the data throughput rates that many applications require. It transfers only 512 bytes once, at most.
- **Long seek time.** Because it use no read-ahead and almost every access needs seek.

Question2: What is the root cause of the problem? In one sentence.

- Because all transfers to the disk are in 512 byte blocks, which can be placed arbitrarily within the data area of the file system and there is no constraints other than available disk place on file growth.

Question3: For each optimization, how does the author get the idea? (Hint: just imagine you're the author. Given the problem, what experiment would you do to get more info?)

- Firstly, in order to improve the throughput rate, I will try to add the block size, which means it can read more content once a time.
- Secondly, however, just adding the block size will cause another serious problem according to a test done by the author, that is when it comes to small files, such large blocks means terrible space waste, so I gonna solve it. But how? What I want to is to try to allowed the division of a single file system block into one or more fragementents so that it can be used to fit the small files. That is exactly what the author try to do.
- Thirdly, to quick the seek time, try to increase the locality of reference to minimize seek time is available. For example, large files can be allocated from the same cylinder so that even larger data transfer are possible before requiring a seek.
- Next, try to use parameterization means we shouldn't ignore the pyhsical characteristics of mass storage devices and by studing the number of milliseconds required to skip over a block and the rate at which the disk spins, it make make the blocks allocated in an optimum configuration-dependent way. First allocate new blocks on same cylinder as previous block. Then let cylinder group summary information keeps count of available blocks in cylinder group at different rotational positions.
- Finally, use layout policy includes global and local policies to increase locality of reference to minimize seek latency and improve layout of data to make large transfer possible.

Question4: For each optimization, does it always work? In which case it cannot work?

- Adding the block size will face the problem when it comes to too many small files.
- Using fragement will cause the problem——disk fragementation, which is really serious when transferring large files.
- The system uses some effective hash algorithm to better locality, but what if the disk is full? If we

confront this case, then the hash will not work effectively because the available room is short.

Question5: Do you have some better ideas to implement a faster FS?

- Improve the physical tech just like promoteing the use of SSD.
- Use an array to record the latest accessing time of these files and if the file hasn't been visited for a long time then try to move it to the lazy position.