The log here is not the function structure log, it’s the meaning of `record`, like a straight thing.

assumption: the main memory will reduce the cost of read, thus disk write will be the main traffic.

solution: log-structured file system with a segment cleaner process continually regenerates empty segments by compressing the live data from heavily fragmented segments.

**Sprite LFS**

## Design for file systems of the 1990’s

technologies: processors, disks, main memory

main memory-> larger, there should be more write rather than read to the disk, memory can serve as the write buffers as well.

wordload: two problems -> small file access and large file access

## Log-structured file systems

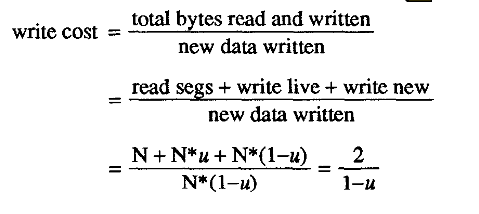
basic idea: buffering a sequence of file changes in the file cache and writes all the changes to the disk in a single write operation.

The basic structure in sprite LFS: inode, but they are stored not in a fixed position, but the logs. There is a \*inode map\* which maintains the locations of inodes.

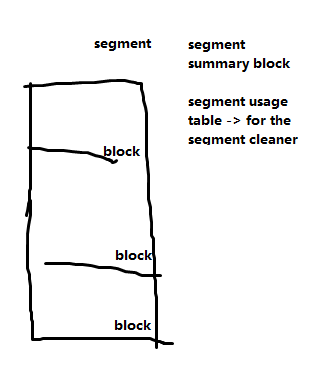
segment cleaner: copy the contents to the memory, write them to a new segment, mark the old segment as new.

policies:

* the time for the cleaner execution(threshold for clean segments)
* the amount of segments a cleaning takes(the threshold for clean segments)
* which segments to be cleaned
* how the live blocks be grouped when these new blocks are written(group blocks which have similar ages into the same block, cleans cold segments at about 75% utilization but waits until hot segments reach a utilization of about 15%.)



u is the utilization of the segment, since if the utilization rate is high, the write cost will be low ⇒ the compact structure of log, most of the segments are nearly full and a few are empty or nearly empty.



## Crash recovery

location of the last disk operation: easy to determine in log, since it’s the end of the log.

two operations:

* checkpoints(30s)
  + position in the log at which all of the file system structures are consistent and complete
* roll-forward
  + scan through the log segments that are written after the last checkpoint.

the interval of checkpoint: trade-off between checkpoints overhead and roll-forward time.

## Experience with the Sprite LFS

\tip: one benefit with log - support versions since we can not delete those garbage files.