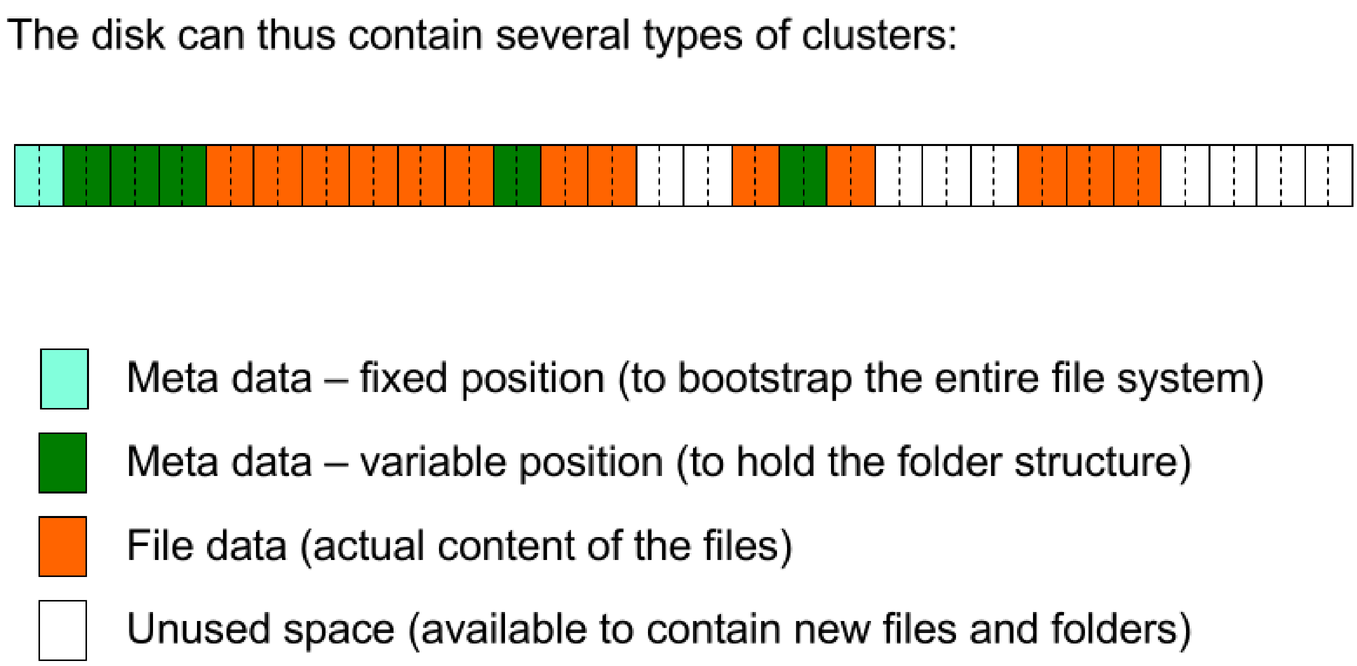
Disk abstraction and HDD

We access files managed by the operating system. The disk is divided in data blocks with an identifier(Logical Block Address). Typically OS groups blocks into clusters to simplify the access to the disk.
The cluster contains:

- File data: actual content of the file
- Meta data: information required to support the file system. They contains files names, directory structures and symbolic links, file size and type, creation, modification, last access dates, security information(owner, access list and encryption), links to the LBA where the file content can be located.



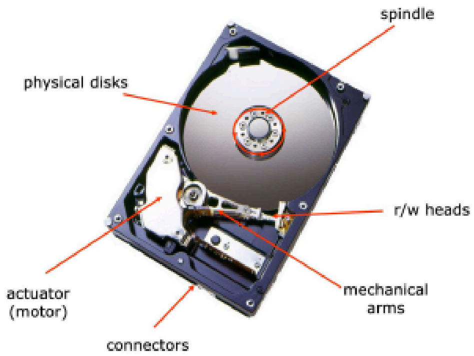
Since the file system can only access clusters, the real occupation of
space on a disk for a file is always a multiple of the cluster size Let us call:

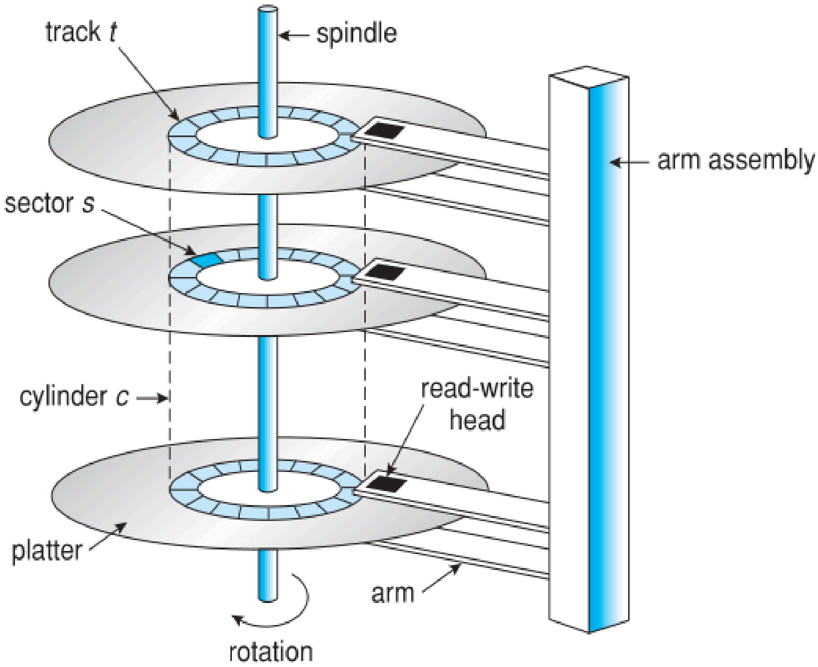
- s: the file size
- c: the cluster size
- a: the actual size on disk
  Then we have: a=ceils(s/c)* c
  And the quantity w = a - s is wasted disk space due to the organization of the file into clusters. This waste of space is called internal fragmentation of files. I am lucky when the file size is equal to the cluster size. The worst case is when the cluster size is smaller by 1 byte.
  When we delete a file we are using logic to indicate in the metadata that the corresponding block can be written on replacing the old data.
  If my continuous blocks cannot contain a file I have to split it in blocks and save it in

different blocks(external fragmentation). This operation slow down the operations so it is needed a cyclic deframmentation(only in HDD on SSD it is crazy).

Hard Drive

A hard disk drive (HDD) is a data storage using rotating disks (platters) coated with magnetic material. Data is read in a random-access manner, meaning individual blocks of data can be stored or retrieved in any order rather than sequentially. An HDD consists of one or more rigid ("hard") rotating disks (platters) with magnetic heads arranged on a moving actuator arm to read and write data to the surfaces.





We have to keep dust out of the disk. In the older model also moving the machine could lead to data loss(expensive laptop had an accelerator that removes the heads from the platter).
The write is the critical aspect of an HDD as the read doesn't risk to corrupt data.

Addressing and Geometry

- Externally, hard drives expose a large number of sectors (blocks)
  - Typically 512 or 4096 bytes
  - Individual sector writes are atomic
  - Have a header and an error correction code
  - Multiple sectors writes may be interrupted (torn write)
    - *Torn writes* happens when only part of a multi-sector update are written successfully to disk
- Drive geometry
  - Sectors arranged into tracks
  - A cylinder is a particular track on multiple platters
  - Tracks arranged in concentric circles on platters
  - A disk may have multiple, double-sided platters
- Drive motor spins the platters at a constant rate
  - Measured in revolutions per minute (RPM)

Currently the technology presents a diameter of 9cm with two surfaces, the rotation speed is 7200-15000 RPM, the track density is 16000 Track Per Inch, heads can be parked close to the center or to the outer diameter mobile drives. Disk buffer Cache is a embedded memory in a hard disk drive that has the function of a buffer between the disk and the computer. Density on a disk remain the same but the size of the disk can increase.
Two different technologies to connect disk to the motherboard:

- Sata
- SCSI(Small Computer Systems Interface):

Types of Delay with Disks

1. Rotational Delay: time to rotate the desired sector to the read head. Related to RPM
2. Seek Delay: time to move the read head to a different track
3. Transfer time: time to read or write bytes
4. Controller Overhead: overhead for the request management

Rotational Delay

Full rotation delay is R = 1/DiskRPM

- In seconds Rsec = 60*R

$T_{rotation\_AVG} = Rsec/2$

Seek Time

Time to move the head to a different track

Several Phases:

- Accelleration
- Coasting (constant speed)
- Decelleration
- Settling

T_{seek} modeling consider a linear dependency with the distance

$T_{seek\_AVG} = T_{seek\_MAX} / 3$

Transfer Time and Controller Overhead

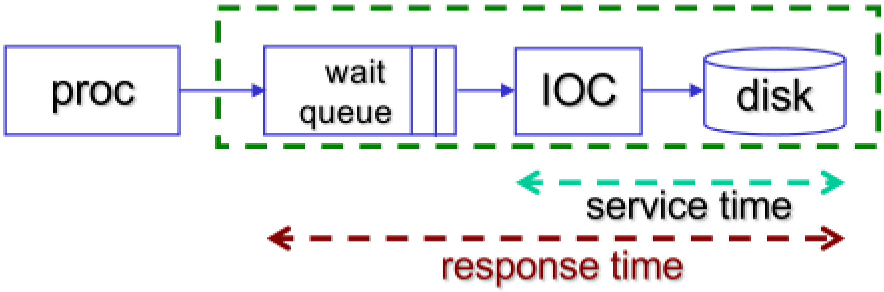
Transfer time

- Final phase of the I/O that takes place
- Time that consider that data is either read from or written to the surface
- Includes the time for the head to pass on the sectors and the I/O transfer
  - rotation speed, storing density

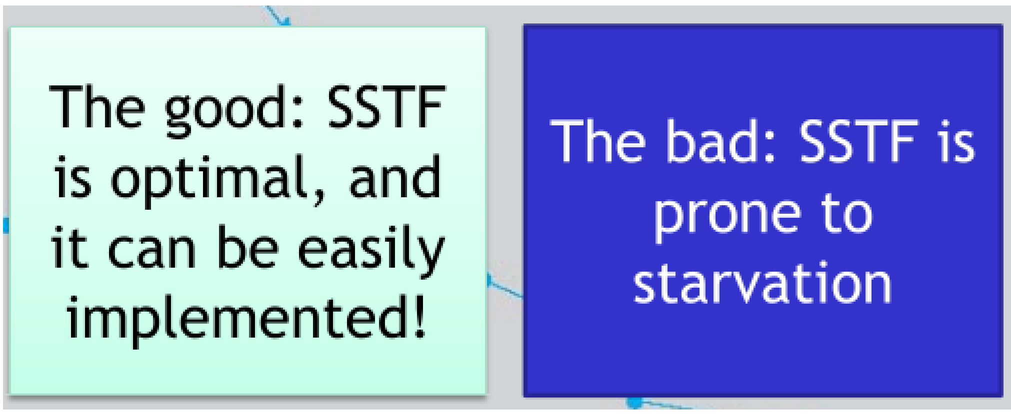
Controller Overhead

- Buffer management (data transfer) and interrupt sending time

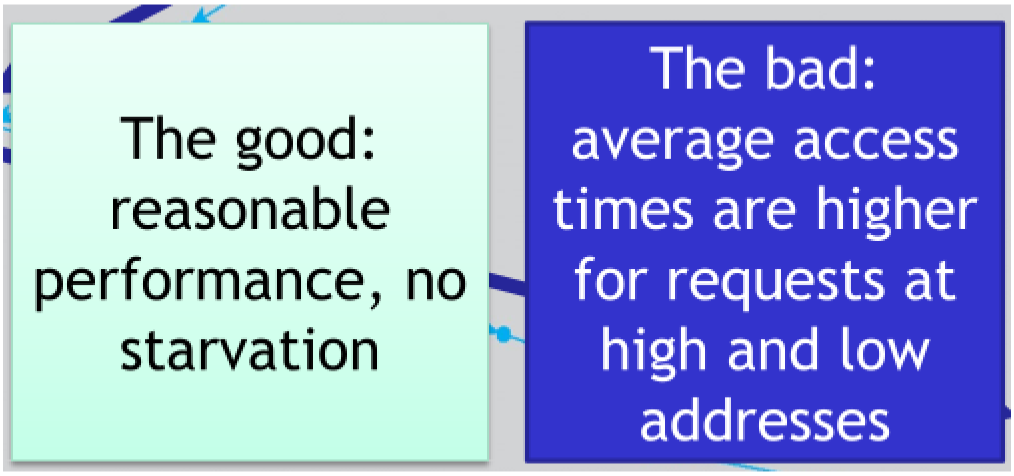
How To Calculate Service Time



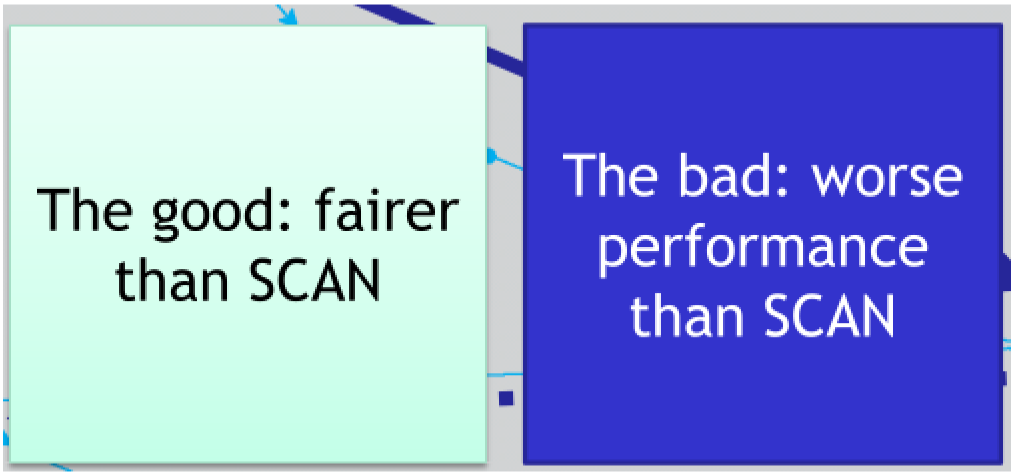
Service time(Single operation time without queueing):
$T_{I/O} = T_{seek} + T_{rotation} + T_{transfer} + T_{overhead}$
Response time($\tilde{R}$): $T_{queue} + T_{I/O}$ where the queue time depends on the queue length, resource utilization, mean and variance of disk service time and request arrival distribution. Having a small service time is really good for your system.
Average Service Time: $T_{I/O} = (1 - DataLocality) * (T_{seek} + T_{rotation}) + T_{transfer} + T_{overhead}$
We want to have another degree of liberty where we want to reduce the seek time. You want to serve the shorter job firstly and the longer job secondly as to maintain a good response rateo. We have to reorder the requests in order of the shortest seek time. It create the problem that the further address are not accesses very frequently, they are always postponed(A scheduler has to be always fair).



So we can implement the so called elevator: we continue to move across the disk to make the target addresses continuous. Problem is arising in power consumption and the addresses in the middle has half the waiting time than the ends.



Circular SCAN tries to solve these problems going in only one direction and swapping the addresses to continue in doing so



Implementing Disk Scheduling

- OS scheduling
  - Requests re-ordering by LBA
  - However, the OS cannot account for rotation delay
- On-disk scheduling
  - Disk knows the exact position of the head and platters
  - Can implement more advanced schedulers
  - But, requires specialized hardware and drivers
- Disk Command Queue
  - Available in all modern disks
  - Queue where a disk stores pending read/write requests
    - Called Native Command Queuing (NCQ)
  - Disk may reorder items in the queue to improve performance
- Joint OS & on-disk scheduling can bring to problems
  - E.g. "NCQ vs. I/O Scheduler: Preventing Unexpected Misbehaviors "

We really want to implement it in the controller and not in the OS to maintain the CPU free to do others, more useful jobs.