

Quando noi lavoriamo su un disco magnetico ogni blocco fisico contiene un blocco logico che noi utilizziamo per esempio all'interno del file system. Ognuno di questi blocchi fisici è sempre leggibile ad ogni istante di tempo: quindi non ci sono dei vincoli particolari con i quali noi dobbiamo andare ad utilizzare questi blocchi per poterli trasferire in RAM, oppure aggiornarli.

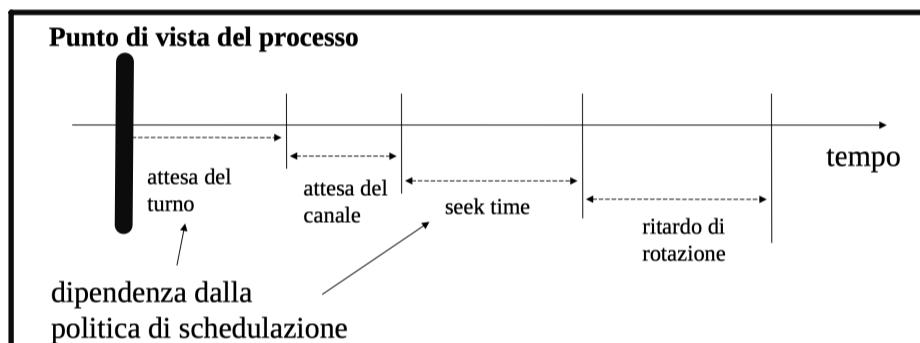
Abbiamo scritto dei nuovi dati all'interno di un file e questi sono finiti in buffer cache - possiamo riportare questo blocco logico all'interno del buffer cache nel corrispettivo blocco fisico che lo ospita - e possiamo fare questa scrittura senza vincoli.

Ad un determinato istante  $T_0$  il software del kernel del sistema operativo ha deciso di prendere dal buffer cache un blocco per andare a scrivere questo blocco sul dispositivo, ad un secondo istante di tempo  $T_1$  andiamo a riprendere il blocco che è stato aggiornato e lo riportiamo esattamente sullo stesso blocco fisico.

- Ogni blocco è leggibile e scrivibile ad ogni istante di tempo
- Non ci sono fasi di vita di un blocco ove date operazioni (ad esempio una scrittura) non possono essere effettuate
- L'usura del dispositivo è essenzialmente legata all'usura delle parti meccaniche
- Non ci sono relazioni dirette tra le operazioni di lettura scrittura dei blocchi e l'usura (ovvero l'usura è primariamente legata al tempo di vita del dispositivo meccanico)
- Le relazioni sono indirette, dovute per esempio al fatto che per posizionare la testina su un blocco questa dovrà essere soggetta ad uno spostamento

## Schedulazione dei dischi magnetici a rotazione

Cerchiamo di capire cosa succede ad un certo punto quando il kernel del sistema operativo decide che è pronta una nuova interazione (ad un determinato istante di tempo) sul dispositivo. Ci sono dei dati che dal buffer cache devono essere scritti sul dispositivo di memoria di massa (disco rigido).



Questa richiesta che c'è stata a questo istante di tempo, perché magari il kernel si è accorto che non c'è più spazio in buffer cache e dobbiamo andare a riportare qualche cosa verso il dispositivo, oppure perché magari il kernel si è accorto che un thread applicativo ha chiamato l'interazione con un file e i dati di quel file non sono in buffer cache, entra in gioco un'interazione con il dispositivo di memoria di massa.

Qui abbiamo la necessità di un'attesa del turno: quindi quella richiesta viene messa in stand by perché ci potrebbero essere altre richieste che devono essere completate prima di lei. Stiamo lavorando secondo un approccio basato sullo scheduling, una pianificazione: stiamo dicendo che questa richiesta che è nata a questo istante di tempo, verrà inoltrata verso il disco, quindi verrà programmato il controller del disco per andare eventualmente a far sì che il disco ci fornisca/acquisisca i dati dalla memoria, che è funzione della politica di scheduling.

Ma per eseguire questa richiesta c'è anche un'attesa del canale: attendere che tutti quei componenti hardware di cui parlavamo prima siano effettivamente liberi per processare questa attività. Internamente allo specifico dispositivo dobbiamo pagare un seek time, ossia il tempo di ricerca della traccia, e poi un ritardo di rotazione, ossia la testina è arrivata sulla traccia, ma il giro della traccia ci deve portare il blocco sotto la testina.

### Parametri

- tempo di ricerca della traccia (seek time)
- ritardo di rotazione per l'accesso al settore sulla traccia

Supponiamo che per muovere una testina noi la dobbiamo muovere di  $n$  tracce. Quindi la testina è correntemente specificata per una specifica traccia e dobbiamo arrivare su un'altra traccia per servire una richiesta e per andare a lavorare sui blocchi a cui siamo interessati. Quindi dobbiamo attraversare una serie di tracce. Supponiamo che nel movimento di questa testina da una traccia sorgente ad una traccia destinazione dobbiamo attraversare  $n$  tracce.  $M$  è una costante che dipende dal dispositivo, e tipicamente in qualche modo è funzione della velocità con cui questa testina attraversa queste tracce.  $S$  è il tempo di avvio quindi è la latenza con la quale il dispositivo eletromechanico impiega per far partire la testina. Il seek time è questo:

### Seek time

- $n$  = tracce da attraversare
  - $m$  = costante dipendente dal dispositivo
  - $s$  = tempo di avvio
- $$\rightarrow T_{\text{seek}} = m \times n + s$$

Una volta che siamo arrivati sulla traccia destinazione il blocco non è detto che stia sotto la testina, quindi dobbiamo attendere la rotazione della traccia affinché questo blocco passi sotto la testina.

### Ritardo di rotazione

- $b$  = bytes da trasferire
- $N$  = numero di bytes per traccia

$$\rightarrow T_{\text{rotazione}} = \frac{b}{r \times N}$$

- $r$  = velocità di rotazione (rev. per min.)

Se noi andiamo a guardare alcuni valori classici di questi parametri, il valore di  $S$  era nell'ordine dei millisecondi. Ma anche.

#### Valori classici

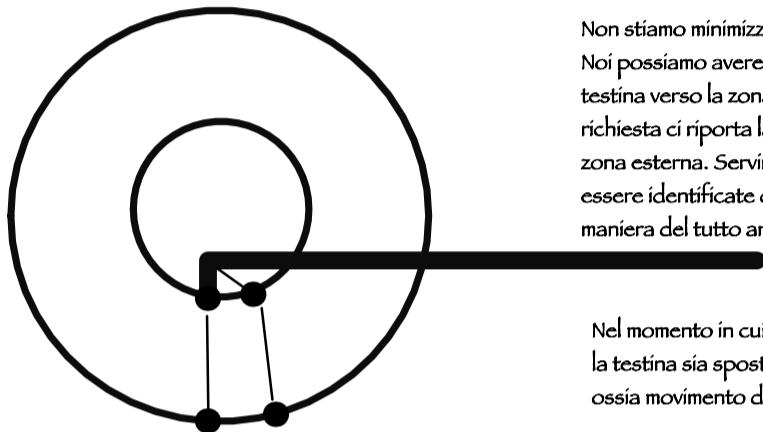
- $s = 20/3/\dots$  msec.
- $m = 0.3/0.1/\dots$  msec.
- $r = 300/3600/\dots/7200$  rpm (floppy vs hard disk)

#### fattore critico

Lo scheduling per andare ad interagire su questi dispositivi hard drive a rotazione, tipicamente era originariamente basato su politiche molto semplici come FCFS.

## Scheduling FCFS (First Come First Served)

- le richieste di I/O vengono servite nell'ordine di arrivo
- non produce starvation
- non minimizza il seek time



Non stiamo minimizzando il seek-time.

Noi possiamo avere una sequenza di richieste in cui, una prima richiesta ci porta la testina verso la zona centrale, una seconda richiesta ce la riporta fuori, una terza richiesta ci riporta la testina nella zona centrale, ed infine una quarta la riporta nella zona esterna. Servire queste richieste esattamente nell'ordine in cui sono arrivate ad essere identificate da "Inoltrare" verso il dispositivo, ci fa muovere questa testina in maniera del tutto arbitraria all'interno del dispositivo di memoria di massa.

Nel momento in cui siamo nella zona centrale, poi per servire la richiesta che richiede che la testina sia spostata nella zona esterna, dobbiamo attendere il passaggio di tracce, ossia movimento della testina.

Andiamo ad analizzare un esempio in cui abbiamo un dispositivo che è un hard drive a rotazione con una traccia iniziale di dove è posizionata la testina, che è la 100esima, e abbiamo una sequenza di tracce che vengono accedute in base alle richieste che sono arrivate attraverso un certo ordine.

### Un esempio

Traccia iniziale = 100

Sequenza delle tracce accedute = 55 - 58 - 39 - 18 - 90 - 160 - 150 - 38 - 184

distanze    45 - 3 - 19 - 21 - 72 - 70 - 10 - 112 - 146

$$\text{lunghezza media di ricerca} = \frac{\sum_{i=1}^{|insiemedist|} dist_i}{|insiemedist|} = 55.3$$