Distributed Video Systems

Chapter 5

Issues in Video Storage and Retrieval

Part I - The Single-Disk Case

Jack Yiu-bun Lee
Department of Information Engineering
The Chinese University of Hong Kong

Contents

5.1 Introduction

- Bandwidth
  - Compressed Video
    - Limited Quality: MPEG4 (~64kbps)
    - Medium Quality: MPEG1 (1~3 Mbps)
    - High Quality: MPEG2 (3 Mbps ~ 12 Mbps)
    - Super-high Quality: MPEG2 HDTV (>10 Mbps)
  - Harddisk
    - SCSI Hard Drive: Transfer Rate ~6MBps (~48Mbps)
  - How many concurrent video streams can be supported?
    - 48Mbps divided by video bit rate?

5.2 Simple Capacity Planning

- Disk Model Revisited

$$T_{seek}(n) = \alpha + \beta\sqrt{n}$$

- n: Number of tracks to seek
- β: Seek-time constant (sec)
- α: Fixed overhead (sec)

$$T_{read}(n) = \alpha + \beta\sqrt{n} + T_{latency} + \frac{Q}{R_{disk}}$$

- Q: Size of data to read (Bytes)
- R_{disk}: Disk transfer rate (Bytes/sec)
- $T_{latency}$: Rotational latency (sec)

How can one obtain these two parameters? (α, β)

5.2 Simple Capacity Planning

Jack Y.B. Lee

- Disk Model Revisited
  - Common disk parameters provided by manufacturer: (Seagate ST12400N SCSI-2)

Disk Parameter	*Value*
Spindle speed	*5411 rpm*
Max latency (r)	*11ms*
Number of tracks	*2621*
Raw transfer rate	*3.35MB/s*
Single-track seek	*1ms*
Max full-stroke seek	*19ms*

$$T_{latency} = \frac{60 \times 1000}{5411} \approx 11$$

Raw transfer rate ← R_{disk}

Single-track seek ← $T_{seek}(1)$

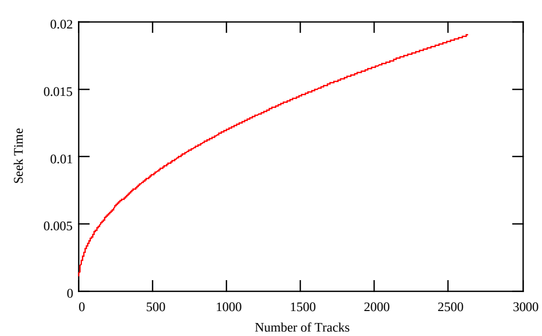
Max full-stroke seek ← $T_{seek}(2620)$

5.2 Simple Capacity Planning

Jack Y.B. Lee

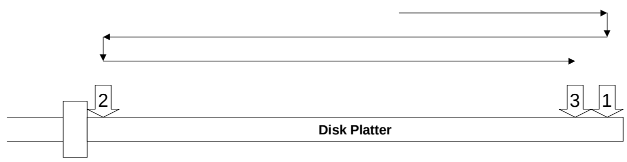
- Disk Model Revisited
  - Solving for α and β:

$$\begin{cases} T_{seek}(1) = \alpha + \beta\sqrt{1} \\ T_{seek}(2620) = \alpha + \beta\sqrt{2620} \end{cases} \Rightarrow \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 6.413 \cdot 10^{-4} \\ 3.587 \cdot 10^{-4} \end{bmatrix}$$



5.2 Simple Capacity Planning

- Disk-Arm Scheduling
  - First-Come-First-Serve (FCFS)
    - Worst-case scenario:

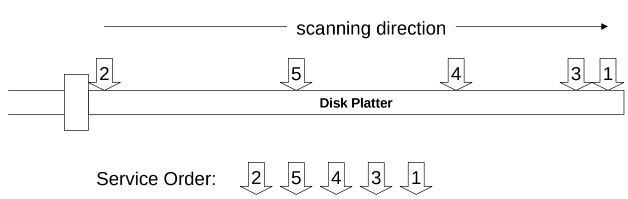


    - Worst-case service time:

$$T_{fcfs} = T_{read}(N_{track} - 1)$$

5.2 Simple Capacity Planning

- Disk-Arm Scheduling
  - SCAN
    - Operations:



    - Length of a service round serving *N* requests:

Head reposition time

$$T_{SCAN} = \sum_{i=0}^{N-1} T_{read}(n_i) + \left(\alpha + \beta\sqrt{N_{track} - 1}\right)$$

Seek distance for request *i*

5.2 Simple Capacity Planning

Jack Y.B. Lee

- Disk-Arm Scheduling
  - SCAN
    - What is the worst-case?
    - **Theorem 5.1**
      - *Given k waiting requests, the worst-case service time with the SCAN algorithm occurs when the k requests are separated by (N_{track}–1)/k tracks (i.e. evenly separated).*
      - *Provable by induction.*
    - Maximum length of a service round:

$$T_{scan}(k) = kT_{read}\left(\frac{N_{track}-1}{k}\right) + \underbrace{\left(\alpha + \beta\sqrt{N_{track}-1}\right)}_{\text{This can be eliminated!}}$$

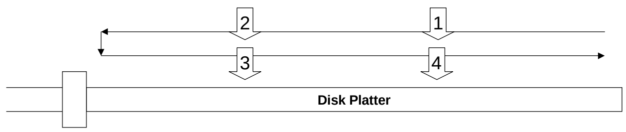
5.2 Simple Capacity Planning

Jack Y.B. Lee

- Disk-Arm Scheduling
  - Circular-SCAN
    - **Theorem 5.2**
      - *Given k waiting requests, the worst-case round time to service all k (k>0) requests in a disk with the C-SCAN algorithm occurs when the requests are separated by (N_{track}–1)/(k+1) tracks.*
      - *Provable by induction.*
    - Maximum length of a service round:

$$T_{cscan}(k) = (k+1)\left(\alpha + \beta\sqrt{\frac{N_{track}-1}{k+1}}\right) + k\left(T_{latency} + \frac{Q}{R_{disk}}\right)$$

- Disk-Arm Scheduling
  - Circular-SCAN
    - Worst-case Scenario:

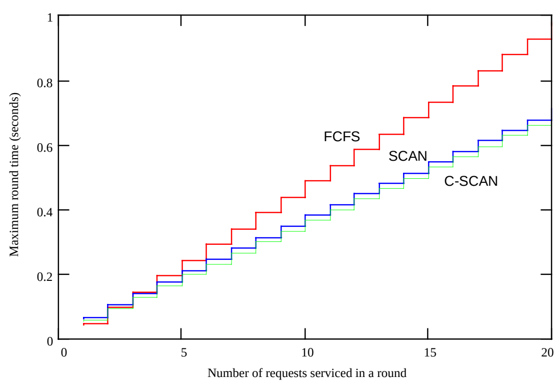


– Some seek time can be saved if the next batch of request is known beforehand.

- Worst-case effective disk throughput:

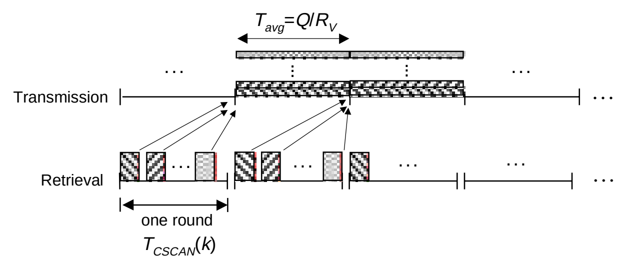
$$S_{cscan}(k) = \frac{kQ}{T_{cscan}(k)}$$

- Comparisons
  - Worst-case round length:
    - Seagate ST12400N (*Q*=65536 bytes)



5.2 Simple Capacity Planning

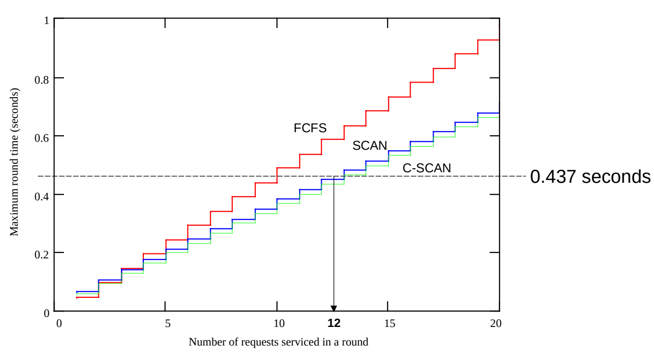
- Comparisons
  - Max. Concurrent Video Streams:
    - Assume video bit-rate = 150KB/s
    - Average time to playback a video block = 64K/150K=0.437 seconds.



For continuity: $T_{CSCAN}(k) \leq T_{avg}$

5.2 Simple Capacity Planning

- Comparisons
  - Capacity
    - Considered only raw disk bandwidth
      - 3.35MBps/150KBps = 22
    - Taking into account of seeking and latency:



5.3 Other Disk Models

- First-Order Approximation
  - Given:
    - Track-to-track seek time
    - Full-stroke seek time
  - Model:

$$T_{seek}(n) = \alpha + \beta\sqrt{n}$$

- Second-Order Approximation
  - Given:
    - Track-to-track seek time, full-stroke seek time and
    - Mean seek time
  - Model:

$$T_{seek}(n) = \alpha + \beta\sqrt{n} + \lambda n$$

5.3 Other Disk Models

- Piecewise Continuous Approximation
  - In real hard drives, seek time is linear except for short ranges.
  - Approximation:

$$T_{seek}(n) = \begin{cases} \overbrace{\alpha_1 + \beta_1\sqrt{n},}^{\text{Non-linear region}} & n < N_L \\ \underbrace{\alpha_2 + \beta_2 n,}_{\text{Linear region}} & \text{otherwise} \end{cases}$$

5.4 Performance Optimization

Jack Y.B. Lee

- Given the disk read function:

$$T_{read}(n) = \alpha + \beta\sqrt{n} + T_{latency} + \frac{Q}{R_{disk}}$$

- How can one increase effective disk throughput?
  - Fixed components:
    - Constant overhead - α
    - Latency - $T_{latency}$
    - Transfer rate - R_{disk}
  - Adjustable components:
    - Seek distance - n
    - Transaction size - Q

5.4 Performance Optimization

Jack Y.B. Lee

- Decreasing the seek distance
  - How?
    - SCAN or C-SCAN
      - Increase the number of requests served in a round.
      - Max. round length:

$$T_{cscan}(k) = (k+1)\left(\alpha + \beta\sqrt{\frac{N_{track}-1}{k+1}}\right) + k\left(T_{latency} + \frac{Q}{R_{disk}}\right)$$

      - Service time per request (under worst-case scenario):

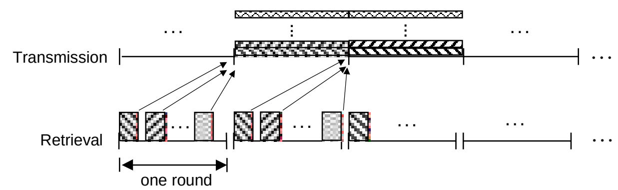
$$\frac{T_{cscan}(k)}{k} = \frac{(k+1)}{k}\left(\alpha + \beta\sqrt{\frac{N_{track}-1}{k+1}}\right) + \left(T_{latency} + \frac{Q}{R_{disk}}\right)$$

    - But can we increase k indefinitely?

5.4 Performance Optimization

Jack Y.B. Lee

- Decreasing the seek distance
  - Tradeoffs
    - Buffer requirement
      - – 2kQ bytes

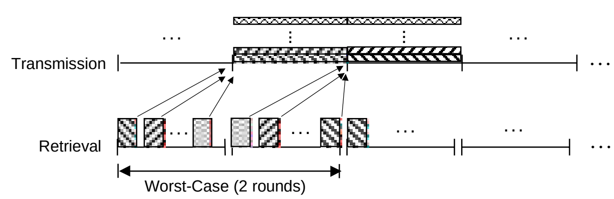


      - – Example
        - Serving 100 requests of each 64KB in a round
        - Buffer requirement is 2x100x64KB=12.8MB

5.4 Performance Optimization

Jack Y.B. Lee

- Decreasing the seek distance
  - Tradeoffs
    - Startup Delay
      - – Two service rounds (worst-case):



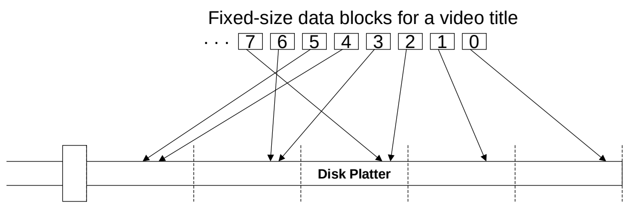
      - – Example
        - Serving 100 requests of each 64KB in a round
        - Startup delay is T_{CSCAN}(100)x2 = 6.628 seconds!

- Increasing Transaction Size *Q*
  - Tradeoffs
    - Buffer requirement
    - Startup delay
  - Practical Considerations
    - Disk sector size
      - Disk reads are performed in whole sectors; i.e. a complete sector is read even if only 1 byte is needed.
      - *Q* should be integral multiples of the disk sector size.
    - Disk read alignment
      - Reads should start at sector boundaries.
    - Memory alignment
      - *Q* should be integral multiples of the memory page size.
      - Buffer memory should be allocated on page boundaries.

- Rotational Latency
  - Problem
    - The worst-case latency depends on rotational speed.
    - The fastest hard drive today spins at 10,000 rpm, which translates into a latency of 6ms.
    - Future hard drives are unlikely to be orders of magnitude faster in spinning.
  - Actually there is a way to reduce the rotational latency.
    - Read the entire track!
    - Maximum latency is then only one sector.
  - There are catches:
    - A track usually is quite large (>1MB), hence buffer requirement and latency becomes large.
    - Tracks could be of different sizes (Section 5.6).

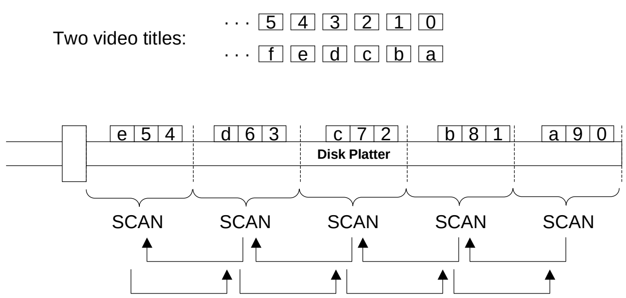
5.5 Internal Striping

- Placement Policy
  - Partition the disk surface into regions
  - Stripe each and every video titles over the regions



5.5 Internal Striping

- Retrieval Scheduling
  - Perform SCAN within a region
  - Disk head moves from region to region in a circular manner



5.5 Internal Striping

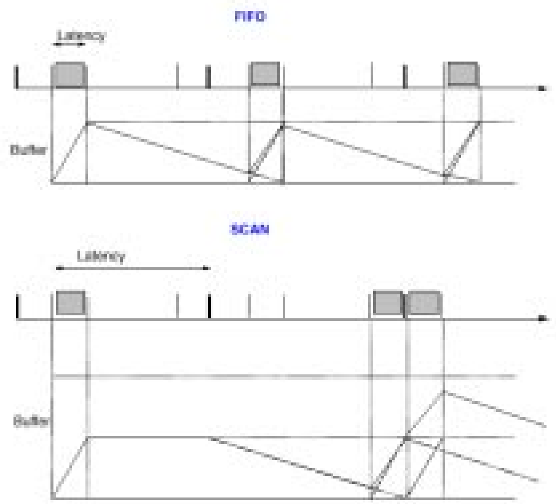
- Comparison with increasing *k* in CSCAN
  - Lower buffer requirement
- Shortcomings
  - Long startup delay
    - All video streams must be synchronized
    - Very large round size
  - Marginal performance gain
    - Depends on seek function
    - Not much gain beyond the non-linear region of the seek-time curve
  - Disk zoning
    - Tracks in real disks could be of different sizes

5.6 Grouped Sweeping Scheme

- Motivation
  - More requests per SCAN, better throughput,
    but longer worst-case delay and buffer requirement.
  - GSS is proposed to stripe balance between these conflicting objectives.
- Principle
  - Divide *n* video streams into *g* groups
  - Streams within a group are served using SCAN
  - Groups are served in a fixed order
- Special Cases
  - If $g=n$ then GSS reduces to FIFO
  - If $g=1$ then GSS reduces to SCAN

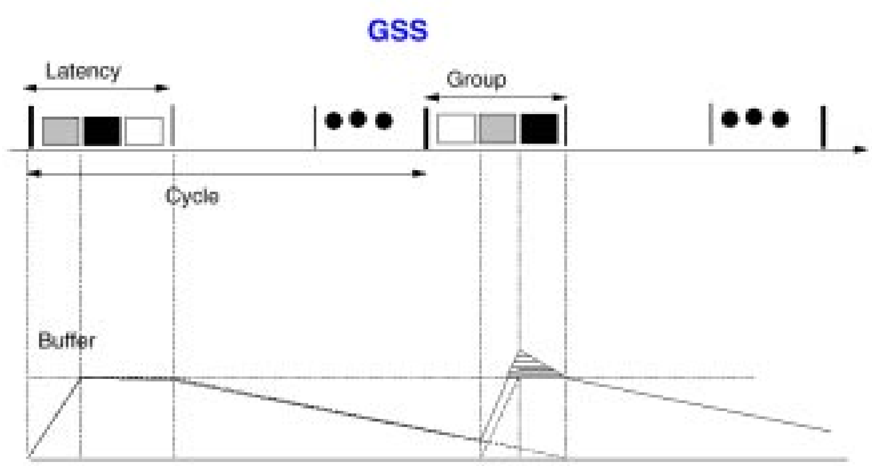
5.6 Grouped Sweeping Scheme

Jack Y.B. Lee



5.6 Grouped Sweeping Scheme

Jack Y.B. Lee



5.6 Grouped Sweeping Scheme

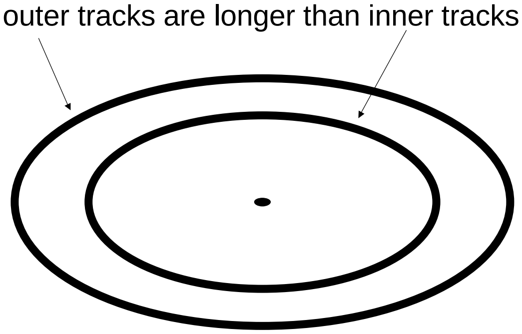
- Buffer Requirement
  - FIFO
    - One buffer per stream (with buffer sharing)
  - SCAN
    - Two buffers per stream
  - GSS

$$B_b = \left(n + \left\lceil \frac{n}{g} \right\rceil \right) k B_m$$

- B_m: Size of a disk block
- k: Number of blocks in one read
- $\left\lceil \frac{n}{g} \right\rceil$: Staging buffers
- n: Playout buffers

5.7 Disk Zoning

- Storage Capacity
  - Hard drive capacity increases rapidly;
  - One technique in achieving this is called *zoning*.
- Principle
  - Rotational speed is constant (CAV)
  - But outer tracks are longer than inner tracks



5.7 Disk Zoning

- Zoning
  - At the same data density (i.e. bytes per inch), the longer the track, the larger the capacity.
  - In practice
    - A disk is divided into multiple *zones*;
    - Tracks within a zone has the same number of sectors.
  - Consequences
    - Tracks can be of different sizes;
    - Transfer rate also depends on the zone.
  - Example
    - Seagate 31200W
      - 23 zones
      - Transfer rates vary from 2.33 to 4.17 MBps

5.7 Disk Zoning

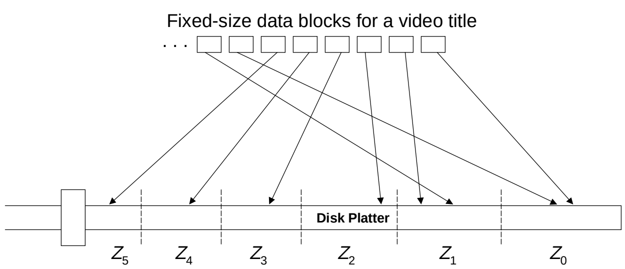
- Implications
  - Effect of zoning on data applications
    - Relatively insignificant
    - Data are not time sensitive
  - Effect of zoning on continuous-media applications
    - Significant!
    - Data are both continuous and time sensitive
- Example: (C-SCAN)

$$T_{cscan}(k) = (k+1)\left(\alpha + \beta\sqrt{\frac{N_{track}-1}{k+1}}\right) + k\left(T_{latency} + \frac{Q}{R_{disk}}\right)$$

What R_{disk} should be?

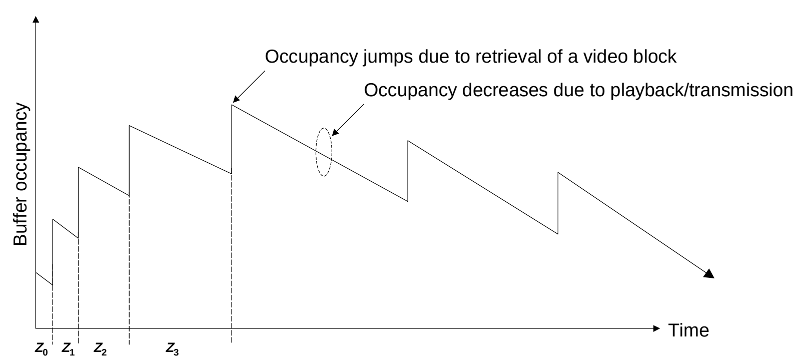
- Simplest Solution
  - Take lowest transfer rate as R_{disk}.
  - Waste disk bandwidth for all except the inner-most zone.
- Solutions with higher effective throughputs?
  - A tradeoff between storage/buffer and throughput
  - Better throughput can be achieved by wasting some storage and using more buffers.
  - Two possible variants: [Ghandeharizadeh 1995]
    - Method 1: Fixed-size blocks
    - Method 2: Variable-size blocks

- Method 1:
  - Placement policy
    - Stripe a video title over all zones using fixed-size blocks in a round-robin manner.

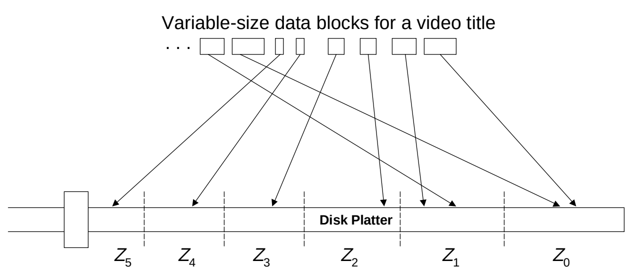


- Method 1:
  - Scheduling policy
    - Given there are *n* zones, a total of *n* data blocks will be retrieved for each video stream in a service round.
    - If there are *m* concurrent streams, a total amount of $2nmQ$ bytes buffer is required.
    - Disk efficiency will probably be high due to the large round size.
  - Drawbacks
    - Both buffer requirement and startup delay will be significantly larger than the case w/o zoning.
    - Storage space will be wasted for all except the inner-most track.

- Method 1:
  - Pipelining
    - Video playback/transmission can start as soon as the first video block has been retrieved.
    - SCAN are used within a zone only (like GSS).



- Method 2:
  - Placement policy
    - Stripe a video title over all zones in a round-robin manner with constant retrieval time (i.e. variable block size).



- Method 2:
  - Scheduling policy
    - Given there are *n* zones, a total of *n* data blocks will be retrieved for each video stream in a service round.
    - If there are *m* concurrent streams, and the block size for zone *i* is u_i, then a total amount of $2m\sum u_i$ bytes buffer is required.
    - Storage wastage is smaller than Method because large blocks are used in outer zones.
  - Drawbacks
    - Buffer management becomes more complicated.
  - Pipelining can again be used to reduce buffer requirement and startup delay.

5.8 Thermal Calibration

- What?
  - In certain hard drives (especially old models), the disk arm positioning must be calibrated periodically to cater for thermal expansion of the hardware.
- So?
  - The drive stops reading/writing while performing a thermal calibration, which can take seconds.
  - This disrupts retrieval schedules in continuous-media applications.
- Solution?
  - While there are ways to take thermal calibration into account, no generally satisfactory way is available.
  - In practice, only drives that do not require thermal calibration should be used in video applications.

5.9 Interactive Viewing Controls

- Basic Interactive Controls
  - Pause/Resume
    - Startup delay is incurred
  - Seeking/skipping/jumping
    - Startup delay is incurred
- Advanced Interactive Control
  - Fast Forward / Rewind (Visual Search)
    - True FF/FR
      - Multiplied bandwidth requirement
    - Data Skipping
      - Difficult to implement on compressed video

5.9 Interactive Viewing Controls

Jack Y.B. Lee

- Separate FF/FR encoded streams

Video title encoded at normal playback rate

Video title encoded at FF playback rate

Video title encoded at FR playback rate

Search Index {Normal Pos, FF Pos, FR Pos}

Data for one video title

<u>User Press FF:</u>
- Pause playback
- Mark Normal Pos
- Find FF Pos based on Normal Pos
- Switch playback to FF stream

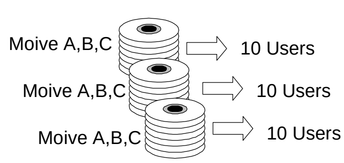
<u>User Release FF:</u>
- Pause playback
- Mark FF Pos
- Find Normal Pos based on FF Pos
- Switch playback to normal stream

- Extra storage is needed but can be reduced by using lower bitrate and/or lower frame rate for FF & FR streams.

5.10 Movie Assignment in Multi-disk Systems

Jack Y.B. Lee

- The Problem
  - One hard drive has limited throughput
  - Replication over multiple hard drives are needed for popular movies
  - But how?
- Full Replication



  - Simple but wastes storage in replicating unpopular movies.

- Full Replication
  - Problems
    - Simple but wastes storage in replicating unpopular movies.
    - What if the size of a movie is larger than the disk storage capacity?
  - Possible Solutions
    - Partial replication
    - Disk striping (next chapter)

- Partial Replication
  - Principle
    - The number of replications should proportional to the popularity of the movie.
  - Movie Popularity
    - Approximation
      - Zipf's distribution $P_i \propto \frac{1}{i^c}$
    - Prediction
      - Based on retrieval patterns in previous days
  - Assignment
    - Construct an optimization model based on resource (storage and bandwidth) and movie popularity (essentially a *bin-packing* problem)
    - Replicating more popular movies and deleting less popular movies

References

- C. Ruemmler, and J. Wilkes, "An Introduction to Disk Drive Modelling," *IEEE Computer*, vol.27, pp.17-28, March 1994.
- P.S. Yu, M.S. Chen, and D.D. Kandlur, "Grouped Sweeping Scheduling for DASD-based Multimedia Storage Management," *ACM Multimedia Systems*, vol.1, pp.99-109, 1993.
- D.J. Gemmell, H.M. Vin, D.D. Kandlur, P.V. Rangan, and L.A. Rowe, "Multimedia Storage Servers: A Tutorial," *IEEE Computer*, vol.28(5), pp.40-9, May 1995.
- T.D.C. Little, and D. Venkatesh, "Popularity-Based Assignment of Movies to Storage Devices in a Video-on-Demand System," *ACM Multimedia Systems*, vol.2(6), pp.280-287, 1995.
- S.D. Stoller, and J.D. DeTreville, "Storage Replication and Layout in Video-on-Demand Servers," *Proc. NOSSDAV'95*, 1995.
- A.N. Mourad, "Issues In the Design of a Storage Server for Video-on-Demand," *ACM Multimedia Systems*, vol(4), pp.70-86, 1996.