

---

# Storage and Retrieval in Continuous-Media Servers

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

---

## Contents

Jack Y.B. Lee

- 1. Introduction
- 2. Simple Capacity Planning
- 3. Other Disk Models
- 4. Performance Optimization
- 5. Internal Striping
- 6. Grouped Sweeping Scheme
- 7. Disk Zoning
- 8. Multi-Disk Servers
- 9. Research Opportunities

## 1. Introduction

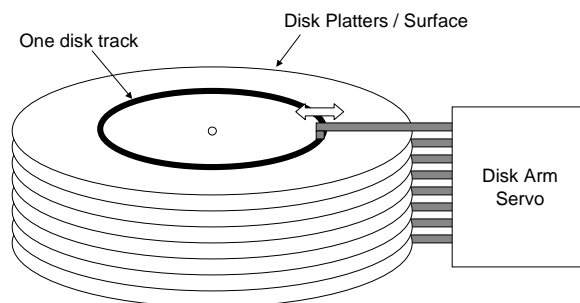
Jack Y.B. Lee

- 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?

## 2. Simple Capacity Planning

Jack Y.B. Lee

- Disk Model



- ◆ The disk platters spin at speed from 3600rpm to 10000rpm;
- ◆ Disk heads in all platters move together.
- ◆ A disk track is further divided into disk sectors.

## 2. Simple Capacity Planning

Jack Y.B. Lee

- Disk Model

- ♦ Fixed Delays

- Processing delay at disk controller;
- Delay at data bus (e.g. SCSI) between disk and controller;
- Head-switching time;

- ♦ Variable Delays

- Rotational Latency
  - Depends on position and spindle speed
- Seek time
  - Depends on number of tracks to seek
- Transfer Time
  - Depends on how much data to transfer to host

## 2. Simple Capacity Planning

Jack Y.B. Lee

- Disk Model

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

Number of tracks to seek  
Seek-time constant (sec)  
Fixed overhead (sec)

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

Size of data to read (Bytes)  
Disk transfer rate (Bytes/sec)  
Rotational latency (sec)

How can one obtain these two parameters?

## 2. Simple Capacity Planning

Jack Y.B. Lee

- Disk Model

- ♦ 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

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

←  $R_{disk}$

←  $T_{seek}(1)$

←  $T_{seek}(2620)$

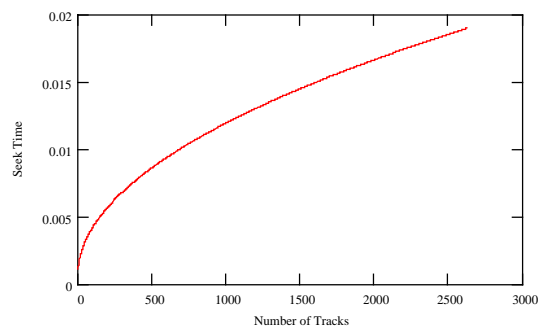
## 2. Simple Capacity Planning

Jack Y.B. Lee

- Disk Model

- ♦ Solving for  $\alpha$  and  $\beta$ :

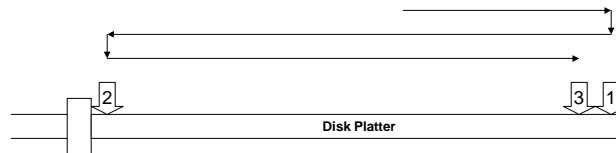
$$\begin{cases} T_{seek}(1) = \alpha + \beta\sqrt{1} \\ T_{seek}(2620) = \alpha + \beta\sqrt{2620} \end{cases} \Rightarrow \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 6.413 \times 10^{-4} \\ 3.587 \times 10^{-4} \end{bmatrix}$$



## 2. Simple Capacity Planning

Jack Y.B. Lee

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



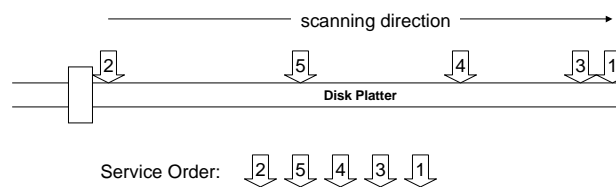
- Worst-case service time:

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

## 2. Simple Capacity Planning

Jack Y.B. Lee

- Disk-Arm Scheduling
  - ♦ SCAN
    - Operations:



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

$$T_{SCAN} = \sum_{i=0}^{N-1} T_{read}(n_i) + \underbrace{(\alpha + \beta \sqrt{N_{track} - 1})}_{\text{Head reposition time}}$$

↑  
Seek distance for request  $i$

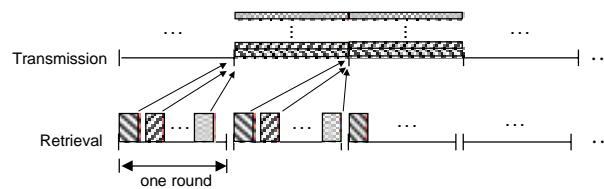
## 2. Simple Capacity Planning

Jack Y.B. Lee

- Disk-Arm Scheduling

- SCAN

- Periodic retrieval of fixed-size data blocks;
- The entire retrieval schedule is known beforehand.
- Read one data block for each video stream in each round.
- Retrievals in a round are serviced using SCAN.



## 2. Simple Capacity Planning

Jack Y.B. Lee

- Disk-Arm Scheduling

- SCAN

- What is the worst-case?
- **Theorem 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).
- Maximum length of a service round:

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

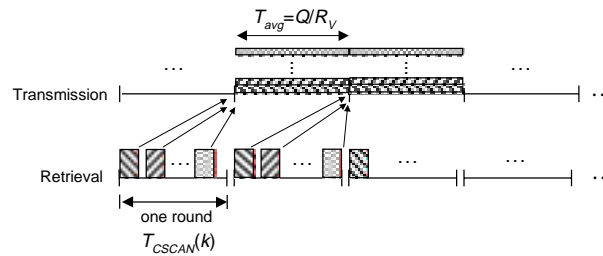
## 2. Simple Capacity Planning

Jack Y.B. Lee

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

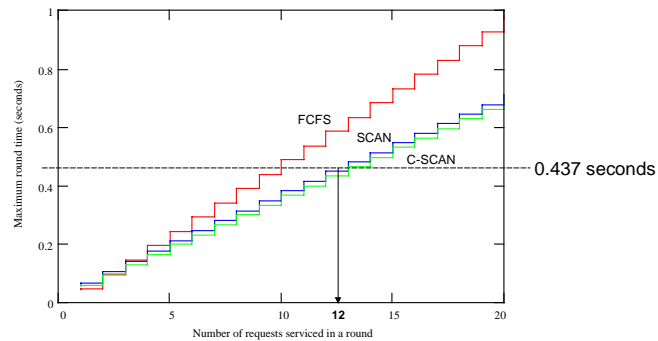
## 2. Simple Capacity Planning

Jack Y.B. Lee

- Comparisons

- Capacity

- Considered only raw disk bandwidth
  - $3.35\text{MBps}/150\text{KBps} = 22$
- Taking into account of seeking and latency:



### 3. Other Disk Models

Jack Y.B. Lee

- 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:

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

### 3. Other Disk Models

Jack Y.B. Lee

- Piecewise Continuous Approximation

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

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

#### 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 -  $\alpha$
  - Latency -  $T_{latency}$
  - Transfer rate -  $R_{disk}$
- ♦ Adjustable components:
  - Seek distance -  $n$
  - Transaction size -  $Q$

#### 4. Performance Optimization

Jack Y.B. Lee

- Reducing the seek distance

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

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

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

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

- But can we increase  $k$  indefinitely?

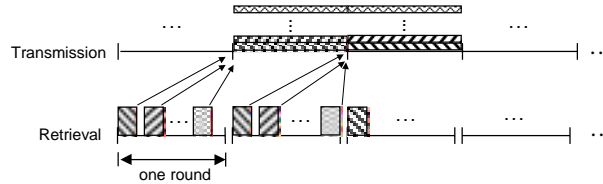
## 4. Performance Optimization

Jack Y.B. Lee

- Reducing the seek distance

- ♦ Tradeoffs

- Buffer requirement
  - $2kQ$  bytes



- Example

- Serving 100 requests of each 64KB in a round
- Buffer requirement is  $2 \times 100 \times 64\text{KB} = 12.8\text{MB}$

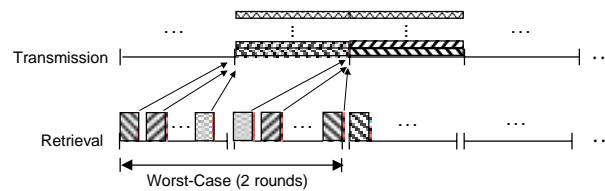
## 4. Performance Optimization

Jack Y.B. Lee

- Reducing the seek distance

- ♦ Tradeoffs

- Startup Delay
  - Two service rounds (worst-case):



- Example

- Serving 100 requests of each 64KB in a round
- Startup delay is  $T_{SCAN}(100) \times 2 = 6.628$  seconds!

## 4. Performance Optimization

Jack Y.B. Lee

- Increasing Transaction Size  $Q$ 
  - ◆ Tradeoffs
    - Buffer requirement
    - Startup delay

## 4. Performance Optimization

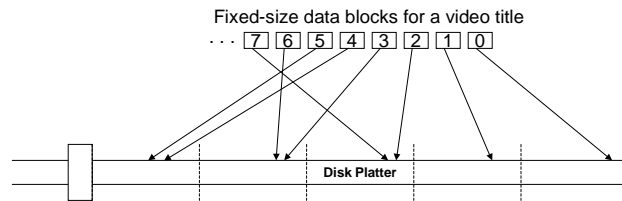
Jack Y.B. Lee

- 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 6).

## 5. Internal Striping

Jack Y.B. Lee

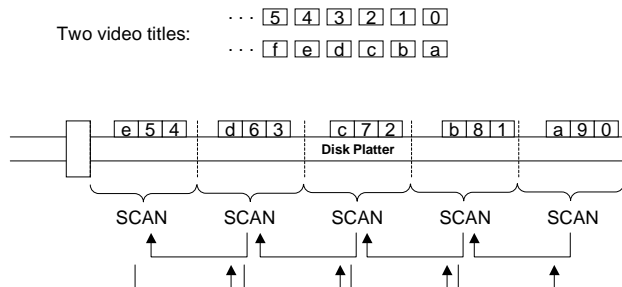
- Placement Policy
  - ◆ Partition the disk surface into regions
  - ◆ Stripe each and every video titles over the regions



## 5. Internal Striping

Jack Y.B. Lee

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



## 5. Internal Striping

Jack Y.B. Lee

- 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

## 6. Grouped Sweeping Scheme

Jack Y.B. Lee

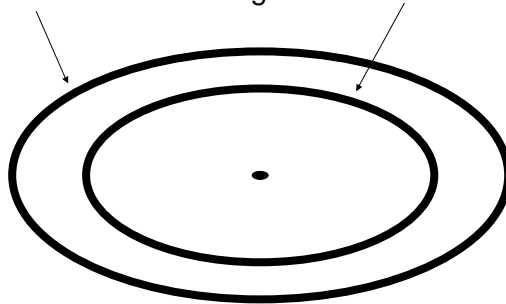
- Motivation
  - ♦ More requests per SCAN, better throughput, but longer worst-case delay and buffer requirement.
  - ♦ GSS is proposed to strike balance between these conflicting objectives.
- Principle
  - ♦ Divide  $n$  video streams into  $g$  groups
  - ♦ Streams within a group are served using 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



## 7. Disk Zoning

Jack Y.B. Lee

- 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



## 7. Disk Zoning

Jack Y.B. Lee

- 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

## 7. Disk Zoning

Jack Y.B. Lee

- 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?

## 7. Disk Zoning

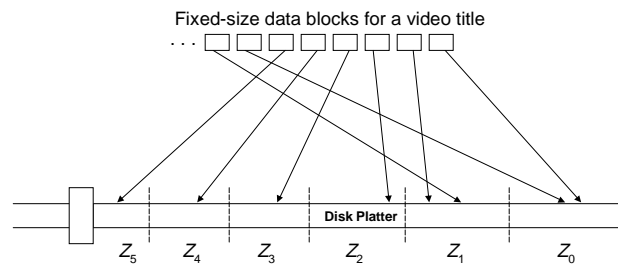
Jack Y.B. Lee

- 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

## 7. Disk Zoning

Jack Y.B. Lee

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



## 7. Disk Zoning

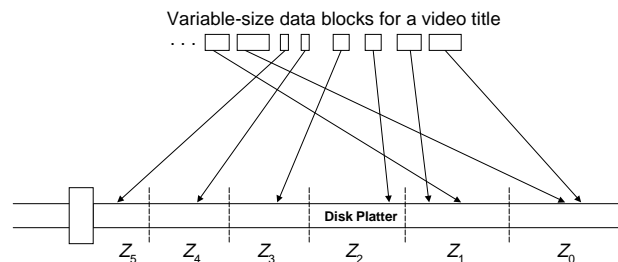
Jack Y.B. Lee

- 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.

## 7. Disk Zoning

Jack Y.B. Lee

- 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).



## 7. Disk Zoning

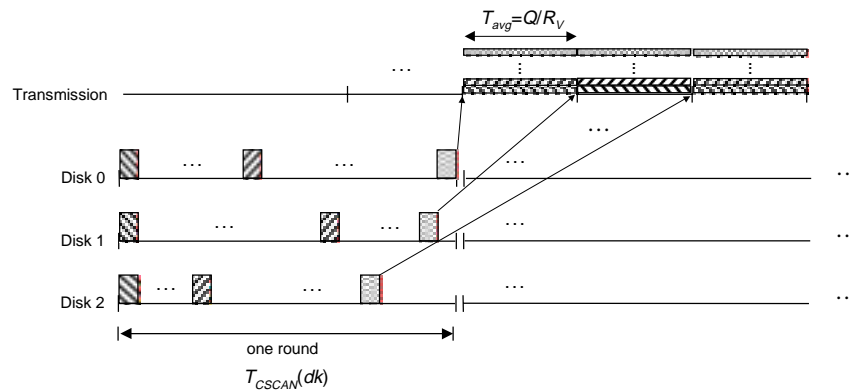
Jack Y.B. Lee

- 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 1 because large blocks are used in outer zones.
  - ♦ Drawbacks
    - Buffer management becomes more complicated.

## 8. Multi-Disk Server

Jack Y.B. Lee

- Concurrent Schedule ( $d=3$  Disks)
  - ♦ SCAN with  $dk$  requests served per round



Storage and Retrieval in Continuous-Media Servers

37

## 8. Multi-Disk Server

Jack Y.B. Lee

- Concurrent Schedule ( $d=3$  Disks)
  - ♦ Performance Gain
    - Higher throughput due to concurrent retrievals;
    - Average load balanced across all disks;
    - Lower seek-time overhead due to more ( $dk$ ) requests served per SCAN round;

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

But in general:  $T_{CSCAN}(dk) \leq dT_{CSCAN}(k)$

Hence it may be possible to serve more requests under the disk-array case than the single-disk case.

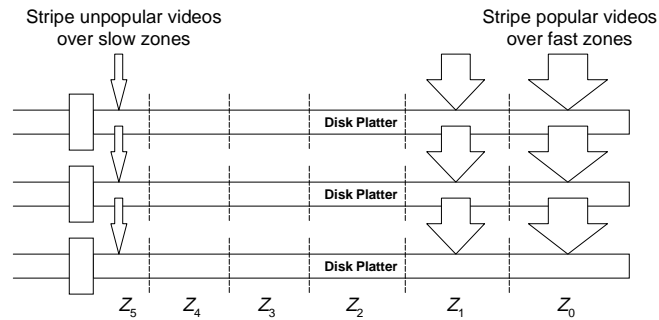
Storage and Retrieval in Continuous-Media Servers

38

## 8. Multi-Disk Server

Jack Y.B. Lee

- Concurrent Schedule ( $d=3$  Disks)
  - ♦ Performance Gain
    - Adapting to Disk Zoning



## 8. Multi-Disk Server

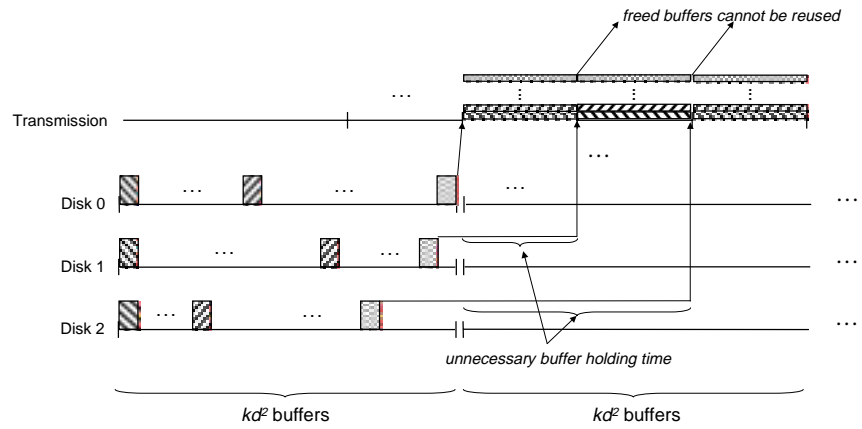
Jack Y.B. Lee

- Concurrent Schedule ( $d=3$  Disks)
  - ♦ Buffer Requirement
    - Double-buffering for SCAN;
    - $dk$  requests served in a round in a disk;
    - total buffers =  $dxdk = 2d^2k$  buffers.
  - ♦ Scalability
    - 64KB stripe units, 20 requests per round;
    - 1 disk - Required buffer per disk = 2.5MB;
    - 8 disks - Required buffer per disk = 20MB!
  - ♦ Problem
    - Scalability is sub-linear because buffer requirement per disk increases for more disks.
    - But why?

## 8. Multi-Disk Server

Jack Y.B. Lee

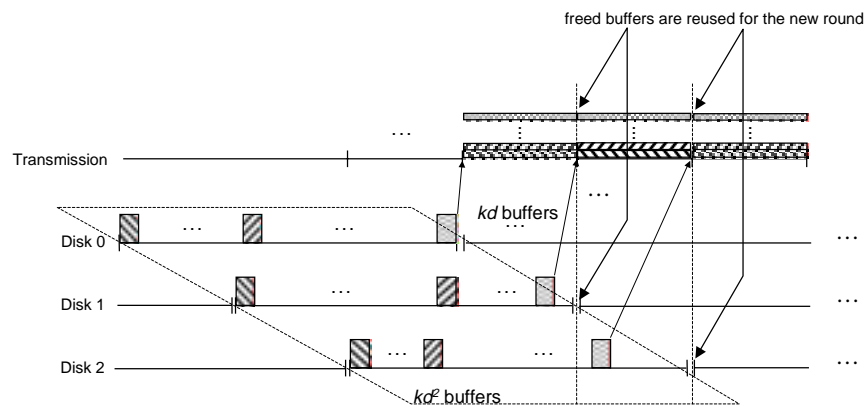
- Concurrent Schedule ( $d=3$  Disks)
  - ♦ Why so many buffers?



## 8. Multi-Disk Server

Jack Y.B. Lee

- Offset Schedule ( $d=3$  Disks)



## 8. Multi-Disk Server

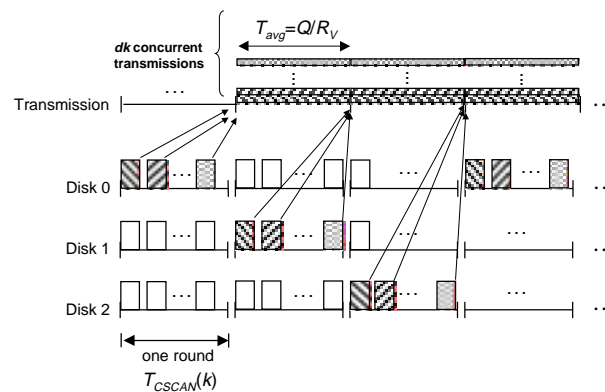
Jack Y.B. Lee

- Offset Schedule ( $d=3$  Disks)
  - ♦ Buffer Requirement
    - total buffers =  $(d+1)dk$  buffers.
  - ♦ Scalability
    - 64KB stripe units, 20 requests per round;
    - 1 disk - Required buffer per disk = 2.5MB;
    - 8 disks - Required buffer per disk = 11.25MB.
  - ♦ Better than concurrent schedule but still not linear!
  - ♦ Anything else we can do?
    - Reduce the number of requests served in a round.

## 8. Multi-Disk Server

Jack Y.B. Lee

- Split Schedule ( $d=3$  Disks)
  - ♦ Serves  $k$  requests per round but different groups of requests in subsequent rounds.



## 8. Multi-Disk Server

Jack Y.B. Lee

- Split Schedule ( $d=3$  Disks)
  - ♦ Buffer Requirement
    - total buffers =  $2dk$  buffers.
  - ♦ Scalability
    - 64KB stripe units, 20 requests per round;
    - 1 disk - Required buffer per disk = 2.5MB;
    - $d$  disks - Required buffer per disk = 2.5MB.
  - ♦ The buffer requirement per disk is finally fixed!
  - ♦ Hence the storage subsystem is scalable to a large number of disks.

## 9. Research Opportunities

Jack Y.B. Lee

- Disk Zoning
  - ♦ Disks with Zoned-Bit-Recording (ZBR) is the norm rather than the exception in practice.
  - ♦ Disk Zoning cripples most existing disk schedulers, including SCAN, GSS, etc.
- Soft-Scheduling versus Hard-Scheduling
  - ♦ Most existing servers are dimensioned using worst-case scenarios. The current disk schedulers provide hard performance guarantees at the expense of efficiency.
  - ♦ Dimensioning with statistical guarantees will likely produce better efficiency.
- Scheduling for Variable-Bit-Rate Media Streams
  - ♦ MPEG2 with constant-quality encoding (e.g. DVD).
  - ♦ MPEG4 with object-based encoding.

## References

Jack Y.B. Lee

- 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 199
- S.D. Stoller, and J.D. DeTreville, "Storage Replication and Layout in Video-on-Demand Servers," *Proc. NOSSDAV'95*, 199
- A.N. Mourad, "Issues In the Design of a Storage Server for Video-on-Demand," *ACM Multimedia Systems*, vol(4), pp.70-86, 1996.