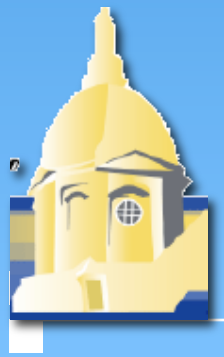
QoS Classes

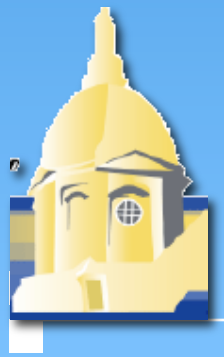
- Guaranteed Service Class
  - QoS guarantees are provided based on deterministic and statistical QoS parameters
- Predictive Service Class
  - QoS parameter values are estimated and based on the past behavior of the service
- Best Effort Service Class
  - There are no guarantees or only partial guarantees are provided



Slides courtesy Prof. Nahrstedt

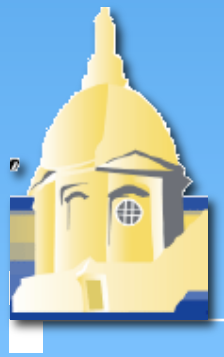
Guaranteed QoS

- We need to provide 100% guarantees for QoS values (hard guarantees) or very close to 100% (soft guarantees)
- Current QoS calculation and resource allocation are based on:
  - Hard upper bounds for imposed workloads
  - Worst case assumptions about system behavior
- Advantages: QoS guarantees are satisfied even in the worst case case (high reliability in guarantees)
- Disadvantage: Over-reservation of resources, hence needless rejection of requests



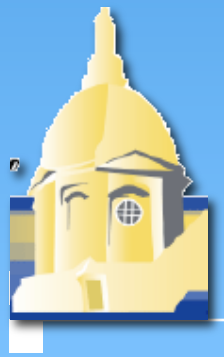
Predictive QoS Parameters

- We utilize QoS values (QoS1, ..QoSi) and compute average
  - QoSbound step at K>i is QoSK = 1/i*∑jQoSj
- We utilize QoS values (QoS1, , QoSi) and compute maximum value
  - QoSK = max j=1,…i (QoSj)
- We utilize QoS values (QoS1, , QoSi) and compute minimum value
  - QoSK = min j=1,…i (QoSj)



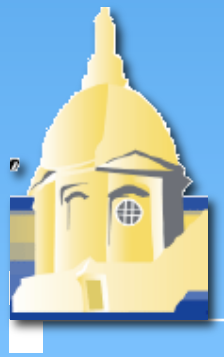
Best Effort QoS

- No QoS bounds or possible very weak QoS bounds
- Advantages: resource capacities can be statistically multiplexed, hence more processing requests can be granted
- Disadvantages: QoS may be temporally violated



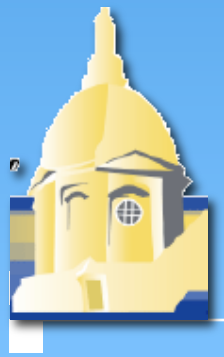
Quality-aware Service Model

- Quality-aware Composite Service
  - Consists of set of autonomous services that are connected into a directed acyclic graph, called service graph
  - Is correct if the inter-service satisfied the following relation:
    - QoSout of Service K ‘satisfies ‘QoSin of Service M iff
    - qKjout = qMlin for qMlin being single QoS value
    - qKjout is in qMlin for qMlin being a range of QoS value
- Example:
  - Video-on-demand service, consists of two services: retrieval service and playback service
    - Output quality of the retrieval service needs to correspond to input quality of playback service, or at least falls into the range of input quality of playback service



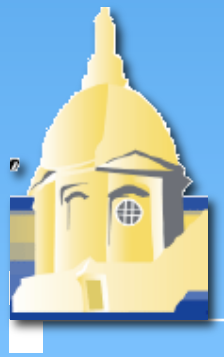
Multimedia Resource Management

- Resource managers with operations and resource management protocols
  - Various operations must be performed by resource managers in order to provide QoS
- Establishment Phase
  - Operations are executed where schedulable units utilizing shared resources must be admitted, reserved and allocated according to QoS requirements
- Enforcement Phase
  - Operations are executed where reservations and allocations must be enforced, and adapted if needed



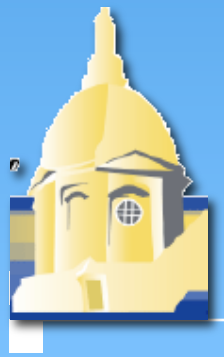
Establishment Phase Operations

- QoS to Resource Mapping
  - Need translation profiles
- Resource Admission
  - Need admission tests to check availability of shared resources
- Resource Reservation
  - Need reservation mechanisms along the end-to-end path to keep information about reservations
- Resource Allocation



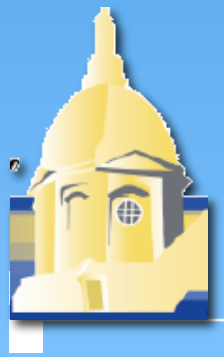
Admission Tests

- Task schedulability tests for CPUs
  - This is done for delay guarantees
- Packet schedulability tests for sharing host interfaces, switches
  - This is done for delay and jitter guarantees
- Spatial tests for buffer allocation
  - This is done for delay and reliability guarantees
- Link bandwidth tests
  - This is done for throughput guarantees



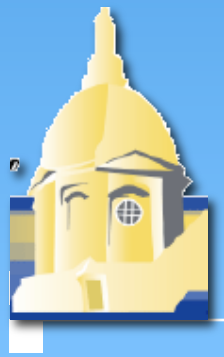
Resource Reservation and Allocation

- Two types of reservations
  - Pessimistic approach - Worst case reservation of resources
  - Optimistic approach - Average case reservation of resources
- To implement resource reservation we need:
  - Resource table
    - to capture information about managed table (e.g., process management PID table)
  - Reservation table
    - to capture reservation information
  - Reservation function
    - to map QoS to resources and operate over reservation table



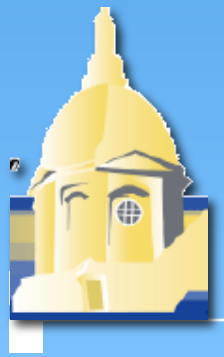
Conclusion – Current State

- Lack of mechanisms to support QoS guarantees
  - Need research in distributed control, monitoring, adaptation and maintenance of QoS mechanisms
- Lack of overall frameworks
  - Need QoS frameworks for heterogeneous environments (diverse networks, diverse devices, diverse OS)



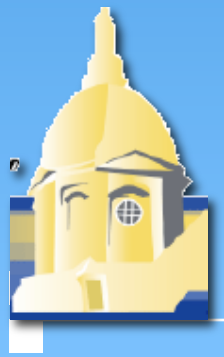
Servers Classification

- Media Servers
  - Push Servers – file servers with streaming model
  - Servers push data towards users
- Traditional File Servers
  - Pull Servers - FTP servers
  - Users pull data in one block at a time by repeatedly calling read to get one block after another
  - RPC Calls



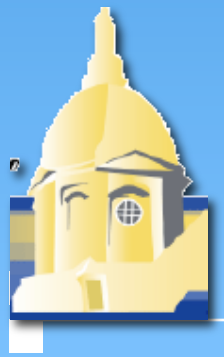
Media Server Requirements

- Real-time storage and retrieval
  - Media quanta must be presented using the same timing sequence with which they were captured
- High-Data Transfer Rate and Large Storage Space
  - HDTV quality: 1280x720 pixels/frame; 24 bits/pixel -> 81 Mbytes per second
  - NTSC quality: 640x480 pixels/frame; 24 bits/pixel ->27MBytes per seconds



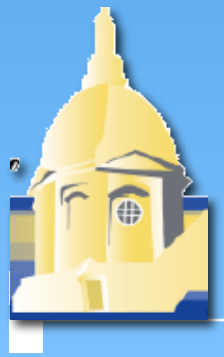
Playback

- Single Stream Playback
  - Possible approach – buffer the whole stream
    - Problem:??
  - Possible approach – prefetch just short video part
    - Problem:
      - Prevent starvation
      - Minimize buffer space requirement
      - Minimize initiation latency
- Multiple Streams Playback
  - Possible approach – dedicate a disk to each stream
    - Problem: ??
  - Possible approach – multiple streams per disk
    - Problems: ??

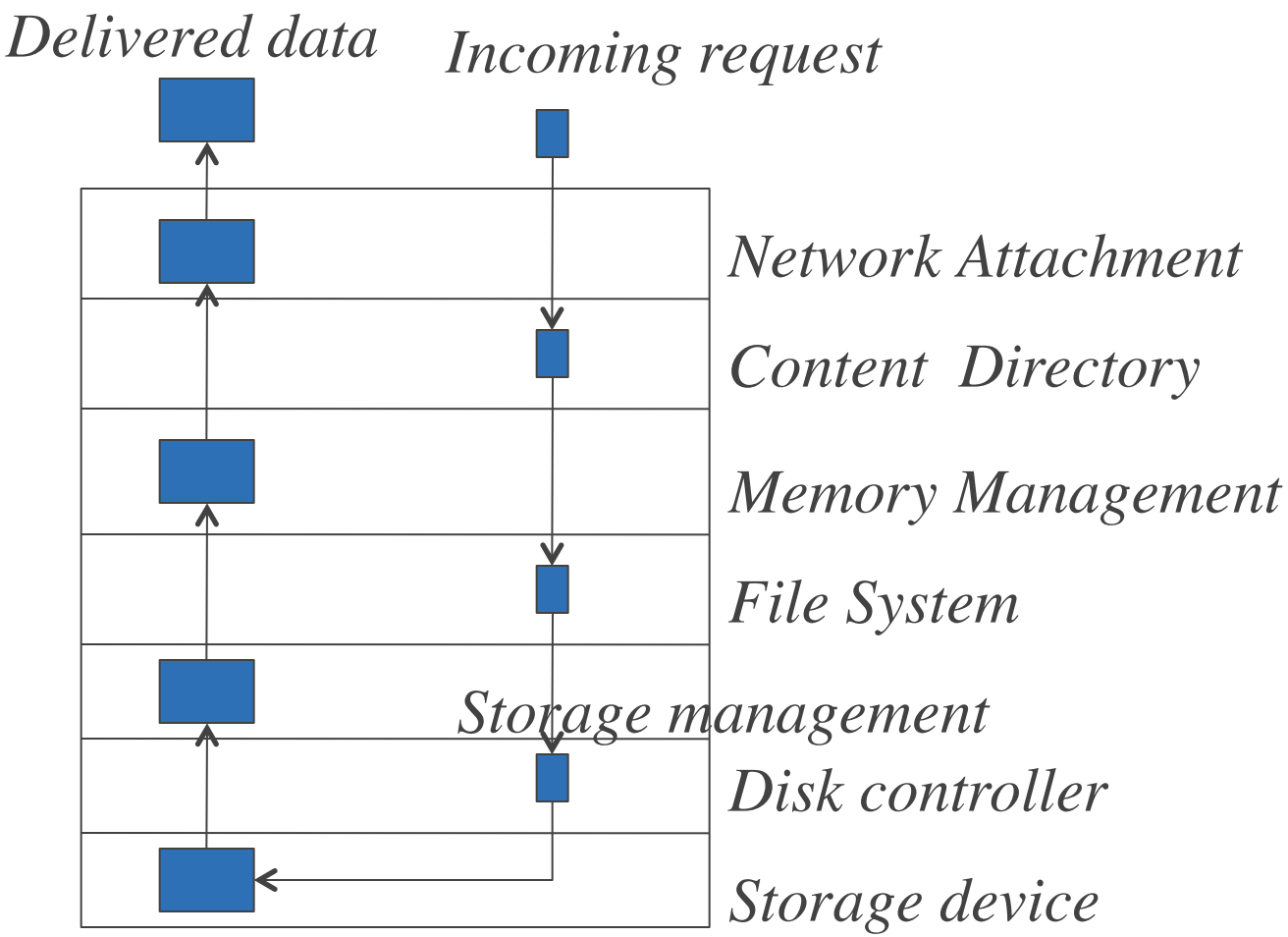


Support for Continuous Media

- Proper management of multimedia disk storage
  - Optimal placement of data blocks on disk
  - Usage of multiple disks
  - Role of tertiary storage
- Admission control
- Special disk scheduling algorithms and sufficient buffers to avoid jitter

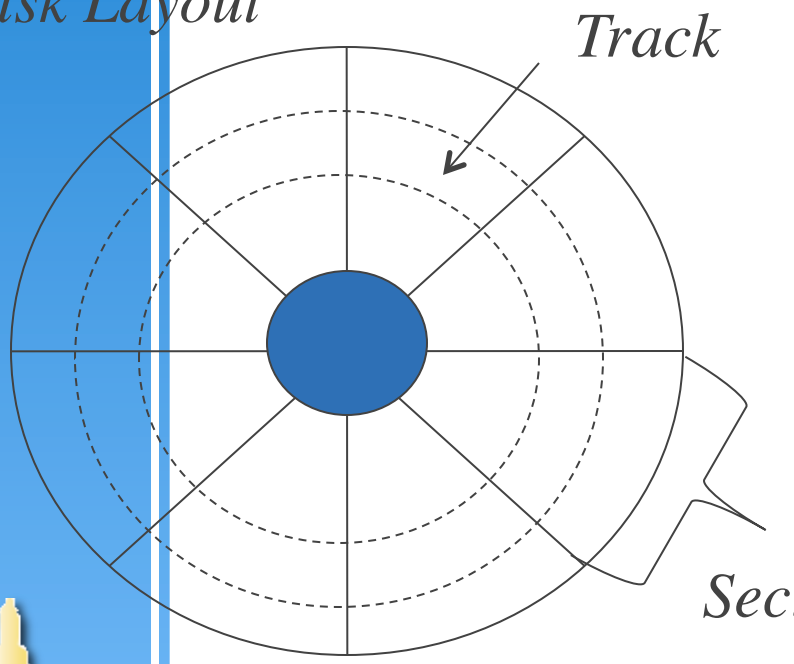


Media Server Architecture



Disk Layout

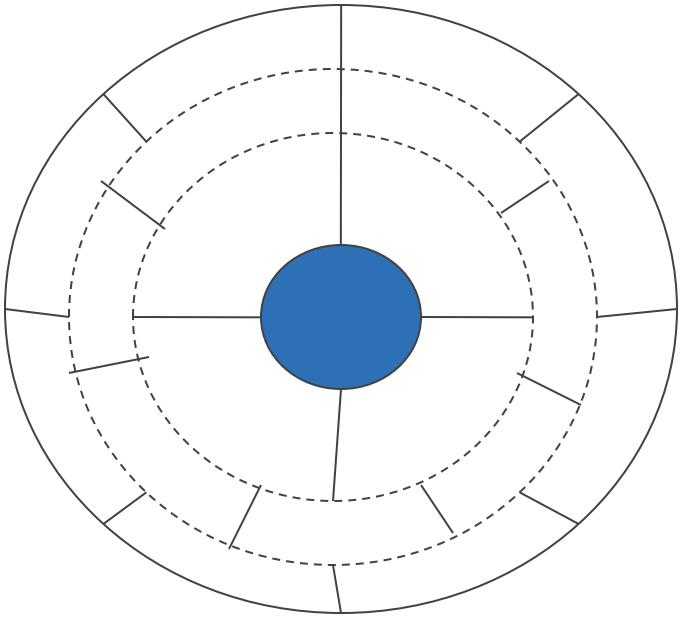
Traditional Random Access Disk Layout



Advantage: Easy mapping of location Information to head movement and disk rotation

Problem: loss of storage space

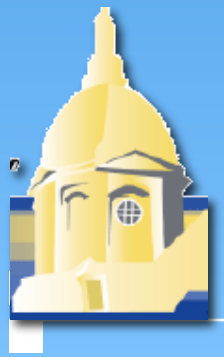
Zoned Disk (ZBR – Zone Bit Recording)



Advantage: Sector size same Rotation speed constant; efficient Usage of space

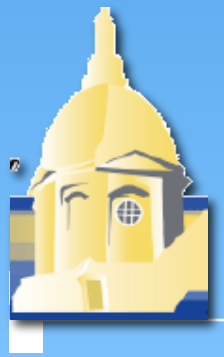
Storage Management

- Storage access time to read/write disk block is determined by 3 components
  - Seek Time
    - Time required for the movement of read/write head
  - Rotational Time (Latency Time)
    - Time during which transfer cannot proceed until the right block or sector rotates under read/write head
  - Data Transfer Time
    - Time needed for data to copy from disk into main memory

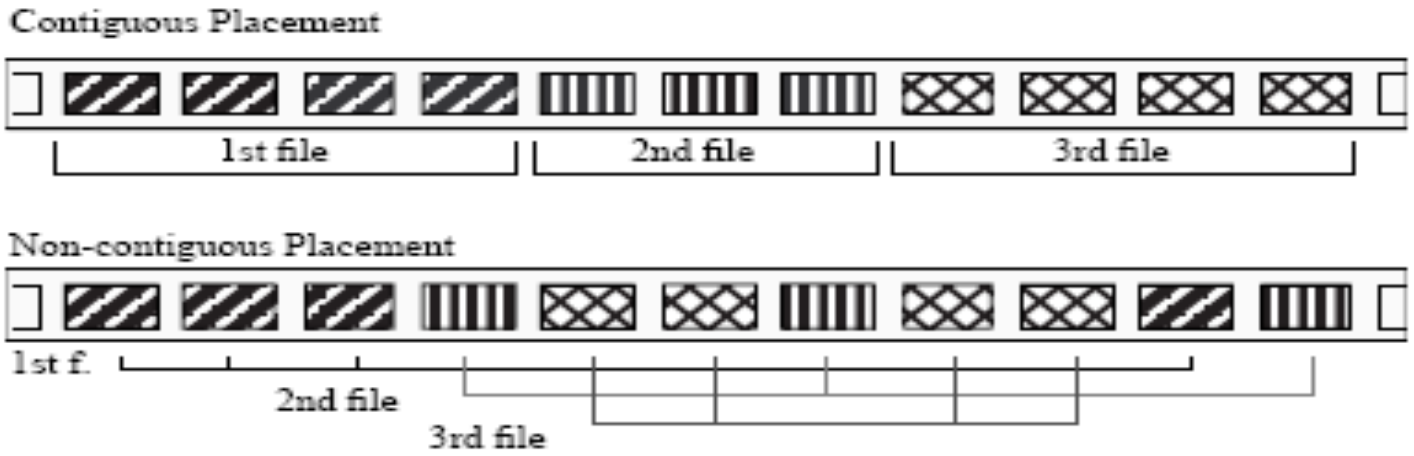


Placement is important

- Disk Layout and Number of disks play important role for media servers
- Data block placement and file placement are crucial for real-time retrieval on media servers



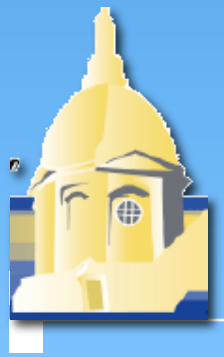
Placement of MM Data Blocks on Single Disk



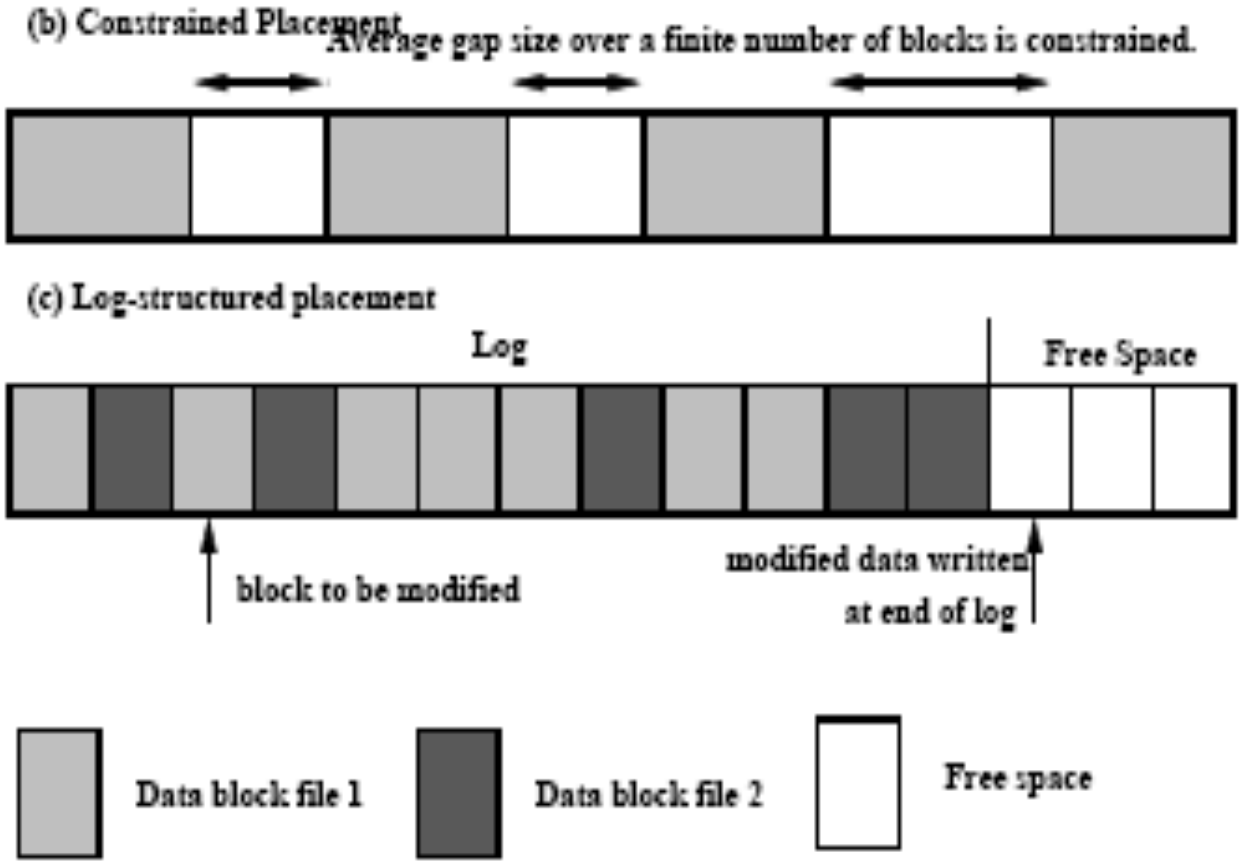
Continuous Placement	Scattered Placement
Simple to implement, but subject to fragmentation	Avoids fragmentation
Enormous copying overhead during insert/delete to maintain continuity	Avoid copying overhead
When reading file, only one seek required to position the disk head at the start of data	When reading file, seek operation incurs for each block , hence **intrafile seek**

Intra-file Seek Time

- Intra-file seek – can be avoided in scattered layout if the amount read from a stream always evenly divides block
- Solution: select sufficient large block and read one block in each round
  - If more than one block is required to prevent starvation prior to next read, deal with intra-file seek
- Solution: constrained placement or log-structure placement

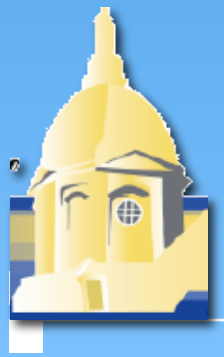


Scattered Non-continuous Placement



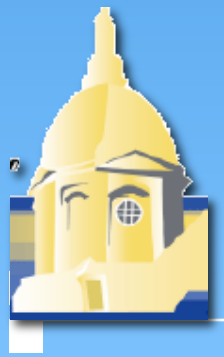
Constrained Placement

- Approach: separation between successive file blocks is bounded
  - Bound on separation – not enforced for each pair of successive blocks, but only on average over finite sequence of blocks
  - Attractive for small block sizes
  - Implementation – expensive
- For constrained latency to yield full benefit, scheduling algorithm must retrieve immediately all blocks for a given stream before switching to another stream



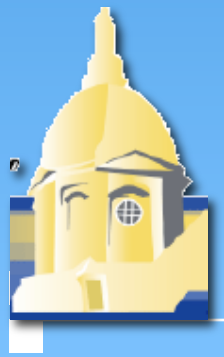
Log-Structure Placement

- This approach writes modified blocks sequentially in a large contiguous space, instead of requiring seek for each block in stream when writing (recording)
  - Reduction of disk seeks
  - Large performance improvements during recording, editing video and audio
- Problem: bad performance during playback
- Implementation: complex



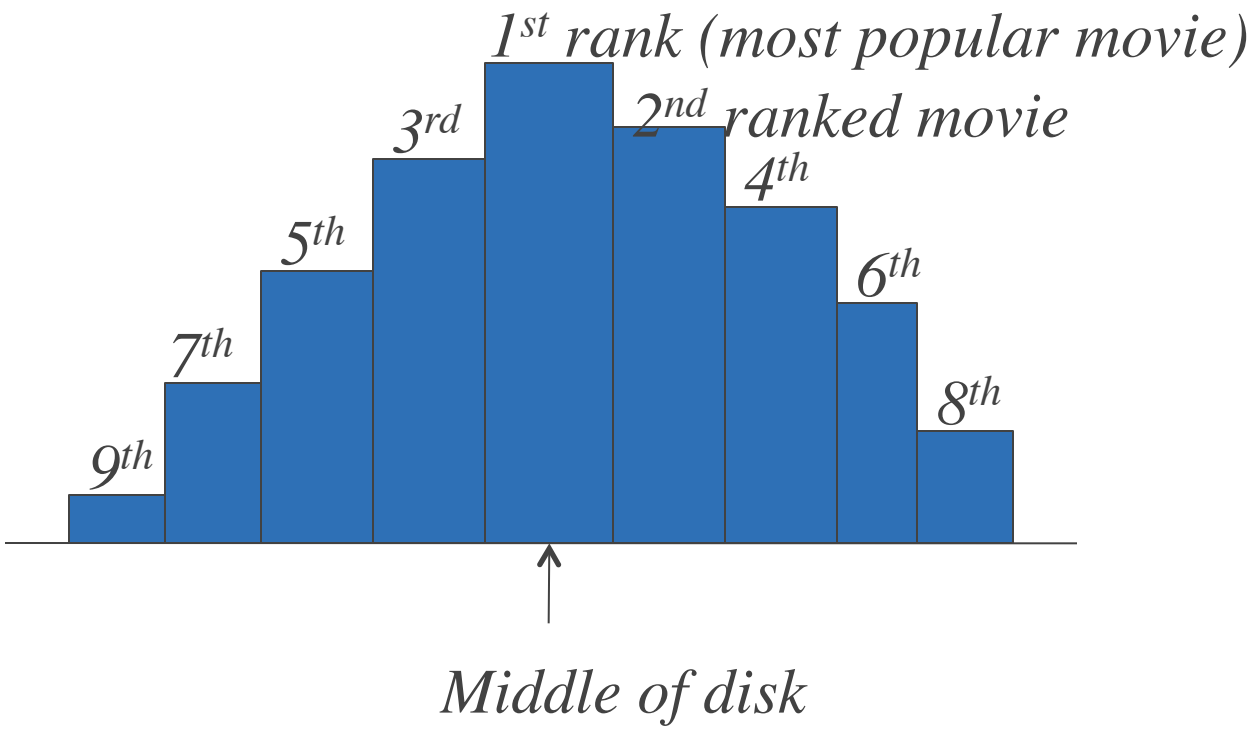
Placement of Multiple MM Files on Single Disk

- Popularity concept among multimedia content - very important
- Take popularity into account when placing movies on disk
- Model of popularity distribution – Zipf's Law
  - Movies are kth ranked
    - if their probability of customer usage is C/k,
      - C = normalization factor
    - Condition holds: C/1 + C/2 + … C/N = 1,
      - N is number of customers

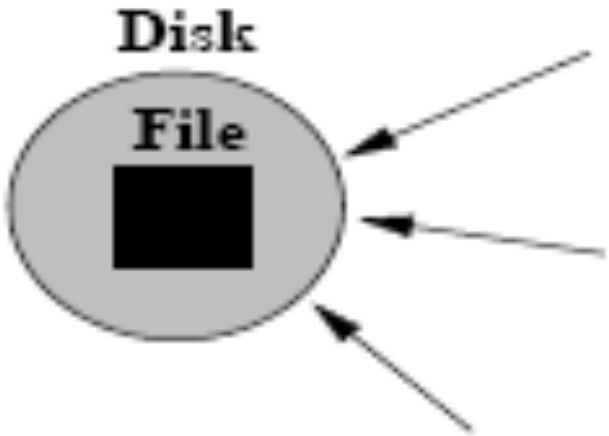


Placement Algorithm for Multiple Files on Single Disk

- Organ-Pipe Algorithms (Grossman and Silverman 1973)

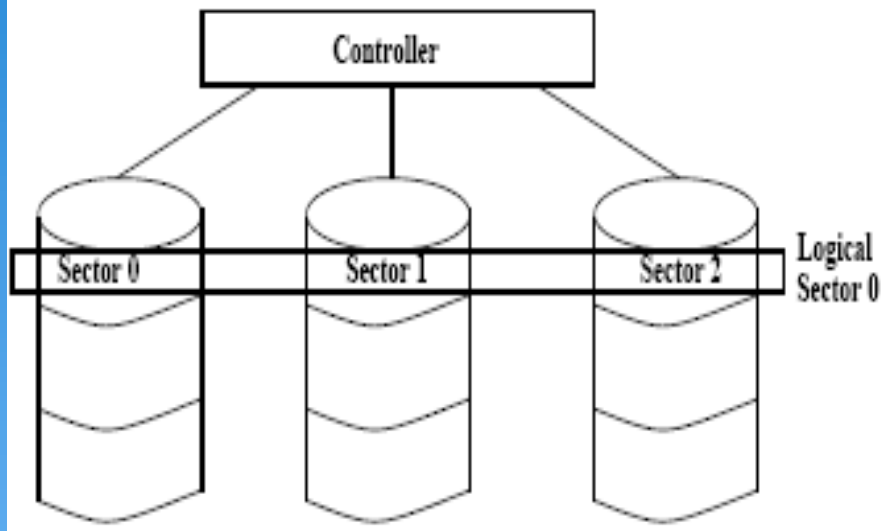


Need for Multiple Disks Solutions for Media Server



- Limitation of Single Disk: Disk Throughput
- Approach: 1 Maintain multiple copies of the same file on different disks
  - Very expensive
- Approach 2: Scatter multimedia file across multiple disks

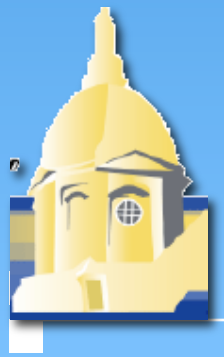
Approach: Data Striping



- RAID (Redundant Arrays of Inexpensive Disks)
  - Addresses both performance and security
  - (0-6) RAID levels – different approach at combining performance enhancements with security/fault-tolerance enhancements
- Disks spindle synchronously
  - Operate in lock-step parallel mode
- Striping improves BW, but does not improve seek or rotational delay

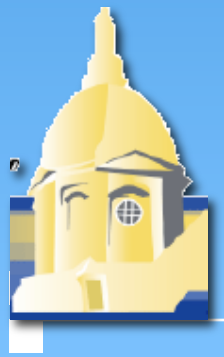
Storage/Disk Management

- Disk access – slow and costly
- Reduce disk access
  - Use block caches (anticipate future reads or writes)
  - Reduce disk arm motion
    - Blocks accesses in sequence (continuously) , place together on one cylinder
    - Interleaved vs non-interleaved storage



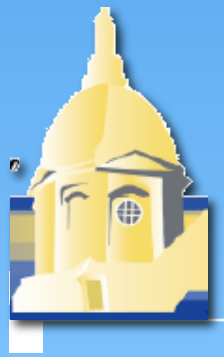
Disk Scheduling Policies

- Goal of Scheduling in Traditional Disk Management
  - Reduce cost of seek time
  - Achieve high throughput
  - Provide fair disk access
- Goal of Scheduling in Multimedia Disk Management
  - Meet deadline of all time-critical tasks
  - Keep necessary buffer requirements low
  - Serve many streams concurrently
  - Find balance between time constraints and efficiency



EDF (Earliest Deadline First) Disk Scheduling

- Each disk block request is tagged with deadline
- Policy:
  - Schedule disk block request with earliest deadline
  - Excessive seek time – high overhead
  - Pure EDF must be adapted or combined with file system strategies



SCAN-EDF Scheduling Algorithm

- Combination of SCAN and EDF algorithms
- Each disk block request tagged with augmented deadline
  - Add to each deadline perturbation
- Policy:
  - SCAN-EDF chooses the earliest deadline
  - If requests with same deadline, then choose request according to scan direction