Typical Applications

- Video on Demand
- Near video on demand
- Internet delivery of news
- Training/travel videos
- Interactive games

ELEN627 Lecture 6

- Typical applications
- System requirements
- Disk scheduling
**Application requirements**

- Continuous retrieval of video, audio

- Service needs to be timely
  - guarantees required

- Guarantees over long periods of time

- Interactive applications require fast response
  - may not be able to prefetch

- Current systems provide best effort service

**Service classes**

- Periodic service
  - CBR
  - VBR

- Interactive service
  - Fast response times

- Regular file service
  - Throughput guarantees may be required
Service components

- Data retrieval from disk
- Transfer of data across system area network
- Processor scheduling for timely response
- Network delivery to user
System level Issues

- Providing Integrated service
- Scheduling and resource allocation
- Adaptability to variations in demand
**Disk or I/O system issues**

- Video has large data sets even after compression
  - 1.5 hour compressed MPEG-1 movie = 1GB
  - How to store this data on disks?

- Video requires high bandwidth
  - 1 MPEG-1 stream needs 200KB/s
  - How to support many streams in a server

- Load balancing
  - Not all streams are equally popular

- Disk scheduling
  - How to provide retrieval guarantees
  - Disk accesses have random seek and latency costs

**Data distribution**

- Use striping
  - Put block 0 on disk 0, block 1 on disk 1, ...
  - Every disk participates in a stream access
  - Load balanced across disks over the life of stream
  - Popular movies can get higher bandwidth

- Put first block of streams on random disks
  - Else, disk 0 will become bottleneck

- Use large blocks, 64KB - 256KB
  - Disk retrieval and striping more efficient
Example

<table>
<thead>
<tr>
<th>Movie/Blocks</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Fig. (a). Movie distribution.

<table>
<thead>
<tr>
<th>Req</th>
<th>Slot</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Disk scheduling

- Disks typically use seek optimizing strategies
  - Reorder requests to reduce overall service time
  - Random seek and rotational costs
  - How to provide retrieval guarantees?

- Real-time scheduling

Fig. (b). Example schedule

- Earliest Deadline First - EDF
  - Seek times can be very high
  - Throughput can be low

- Can we do better?
Disk scheduling (cont’d)

- Group requests into rounds or batches
- Make all the requests in a round have deadlines at the end of the round
- Now all requests in a round can be seek optimized
- As long as total service time for round ≤ length of round, we are ok
- Combines seek optimization with EDF, SCAN-EDF

Performance of SCAN-EDF
Integrated Disk Service

- Deterministic Guarantees for VBR streams
- Fast response time for Interactive requests
- Throughput guarantees for file requests

Aperiodic response of SCAN-EDF
Guarantees for VBR streams

- Demand of application changing over time
- Could use peak demands
  - Provides low throughput
- Use Demand traces
  - Compute how much data needed in each round
- Maintain Load trace on the system
- Combine demand trace and load trace to see if stream can be admitted
- Can provide better throughput
Scheduling for Integrated service

- Loop1: Combine and SCAN order periodic and aperiodic requests

- Make equal sized subgroups

- Serve Interactive requests in FIFO order

- Loop: Pick the first interactive request (if any)

- Group it with closest subgroup

- Serve that bunch

- Go to Loop if not end of round, else go to Loop1
Disk utilizations

Impact of aperiodic requests