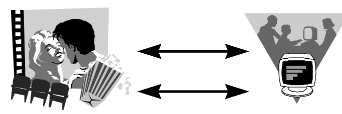
Designing File Systems for Digital Video and Audio



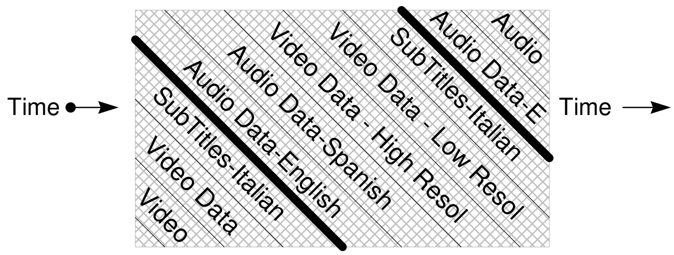
P. Venkat Rangan
and Harrick M. Vin

Purpose and Relevance of Paper

- This paper analyzes specific requirements of multimedia file systems such as
  - media storage and retrieval
  - synchronization between multiple data streams
  - requirements and relations for ensuring continuity of display and retrieval of data
  - bounds for scattering parameter and methods for maintaining these bounds
  - admission control requirements for concurrent requests by users to multimedia data storage bandwidth
- This paper also defines abstractions for conceptualizing multimedia storage and retrieval, such as <u>ropes</u> and <u>strands</u>, and describes how editing of these strands may be accomplished

Relevant Terminology

- **Strand** - an immutable sequence of continuously recorded media samples, such as *video*, *audio*, and *captioning information*
- **Multimedia Rope** - a collection of individual *multimedia strands* tied together by synchronization information



Multimedia "Rope" With Associated Strands

Relevant Terminology -Continued

- **Frame** - Basic video unit
- **Sample** - Basic audio unit
- **Block**- Basic unit of disk storage, which is either homogeneous (e.g. contains only video data) or heterogeneous (e.g. contains both audio, video and textual data)
- **Granularity** - size of storage blocks used on disk
- **Scattering Parameter** - the separation between successive blocks of storage
- **Continuity Requirement** - media information must be available at the display device at or before the time of its playback
- **Synchronized Playback Requirement** - strand playback must occur at exactly the same rate at which it was recorded

Fundamental Requirements

- Audio, Video and Textual Information are separated *at the input* and arrive at the file system as three different streams
- These three components will be routed to three different output devices
- These components may either be stored *together*, which requires processing and merging three streams, or *separately*, which requires the file system to explicitly maintain *temporal relationships* among the media to ensure synchronization during retrieval
- Playback and Storage occur in real-time, so the file system must organize data so that rapid retrieval is possible
- Large File Size- System must deal with this so as to minimize copying of data on the disk

Satisfying the Continuity Requirement

Sequential Architectures

Serialized Read and Display (and Capture/Store) operations; the sum of the time to read a block from disk and the time to display it must not exceed the duration of its playback

Scattering Parameter-sec
Video Storage Granularity-frames/block
Size of Video Frame-bits/frame

Architecture assumes one buffer associated with the video device

$$\left(l_{ds} + \frac{\eta_{vs} * s_{vf}}{R_{dr}}\right) + \frac{\eta_{vs} * s_{vf}}{R_{vd}} \le \frac{\eta_{vs}}{R_{vr}}$$ seconds/block

Video Recording Rate - frames/sec
Rate of video display - bits/sec
Rate of data transfer from disk - bits/sec

Satisfying the Continuity Requirement

Pipelined Architectures

Read and Display operations are performed in parallel; assuming one buffer for block being transferred and another for block being displayed, time to read a block must not exceed the duration of its playback

Scattering Parameter-sec

Video Storage Granularity-frames/block

Size of Video Frame-bits/frame

$$l_{ds} + \frac{\eta_{vs} * s_{vf}}{R_{dr}} \le \frac{\eta_{vs}}{R_{vr}} \quad \text{seconds/block}$$

Video Recording Rate - frames/sec

Rate of data transfer from disk - bits/sec

Architecture assumes <u>two buffers</u> associated with the video device (one for data transfer and one for block display)

Satisfying the Continuity Requirement

Concurrent Architectures

Up to *p* concurrent disk accesses allowed; time to read a block must not exceed the duration for playback of (*p* - 1) blocks

Scattering Parameter-sec

Video Storage Granularity-frames/block

Size of Video Frame-bits/frame

$$l_{ds} + \frac{\eta_{vs} * s_{vf}}{R_{dr}} \le (p-1) * \frac{\eta_{vs}}{R_{vr}} \quad \text{seconds/block}$$

Degree of Concurrency

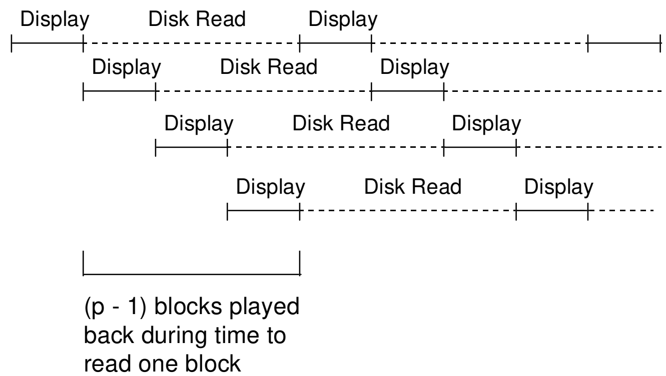
Video Recording Rate - frames/sec

Rate of data transfer from disk - bits/sec

Architecture assumes <u>*p* buffers</u> associated with the video device (one for each concurrent block transfer))

Satisfying the Continuity Requirement

Concurrent Architectures - Example for p=4



Satisfying the Synchronized Playback Requirement

- Forced Synchronization - Uses a clocking device; display process must wait until specific time for displaying each block
- Automatic Synchronization - Force effective access time per media block to be equal to the playback duration of that block

Discussion

- Continuity Requirement may be satisfied on average by introducing anti-jitter delay at the beginning of request (e.g. by performing read-ahead of media blocks prior to sending previously acquired data to display device)
- When Continuity Requirements are satisfied for k successive blocks of a strand, read-ahead of *k* blocks is required in order to guarantee that the next group of *k* blocks can be retrieved from the disk in the time required to display the previous *k* blocks of the strand (*pk* for concurrent architectures).

k blocks displayed; continuity requirements satisfied

k blocks read-ahead (next group for display)

Discussion (Continued)

- "VCR-Functions" - Functions such as "fast-forwarding" may be supported by supporting continuity requirements at the fastest required display rate; however, supporting "slow motion" may result in continuity requirements being over-satisfied, with accumulation of media blocks in buffers
  Solution: Dynamically reconfigure disk to switch to another task when buffers are full, and switch back when buffers empty

Discussion (Continued)

- Storage of Multiple Media Strands (Homogeneous Blocks) - Here, if we assume that audio data is stored much more compactly than video data, such that the same storage block used to hold one block of video data holds enough audio data for *n* blocks of video display. The continuity requirement then becomes

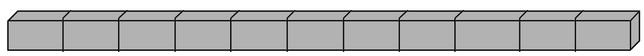
$$\frac{1}{n}\left(\underbrace{l_{ds}^{a} + \frac{\eta_{as} * s_{as}}{R_{dr}}}_{\text{Audio}} + \underbrace{\sum_{i=1}^{n} l_{ds}^{v} + \frac{\eta_{vs} * s_{vf}}{R_{dr}}}_{\text{Video}}\right) \le \frac{\eta_{vs}}{R_{vr}} \text{ seconds/block}$$

Discussion (Continued)

- Transfer of Media Blocks Through Memory to Display Device -
  - More suitable for heterogeneous blocks (due to requirement for additional processing necessitated by storing >1 media information in a storage block)
  - Requires bandwidth for retrieval from storage <u>and</u> routing to display device buffer(s)
- Transfer of Media Blocks *Directly* to Display Device -
  - Requires half internal device bandwidth
  - Device buffer sizes determine granularity of storage required
  - This approach is considered preferable due to lower hardware requirements

Servicing Multiple Requests

- The system can only accept a limited number of requests without violating the continuity requirements of any of the requests
- File system performs “time division multiplexing” such that during a given <u>round</u> multiple block transfer requests are serviced



time slices within Round i dedicated
for servicing media block transfers

Servicing Multiple Requests (Continued)

- Since the continuity requirements assume playback at the fastest attainable display rate, display at a lower rate may lead to accumulation of data in the display subsystems of the various requests
- This accumulation is controlled by dynamically regulating the number of data blocks transferred to each request per round

Talk By Jeff Waldrop - For CS 8113D Multimedia

From Media Strands to Multimedia Ropes

- Media strands are tied together by inter-media synchronization to form a multimedia rope
- Rope contains

- Creator Name	- Rate of Recording
- Rope Length	- Granularity of Storage
- Access Rights for Rope	- Block-Level Correspondence
- Unique ID for each strand	(for synchronizing playback start)

- Block-Level Corresp + Rate of Recording ⇨ inter-media synch
- Silence Elimination - audio data <u>eliminated</u> after silence detection with explicit delay holders placed in audio strands

Implementation of Prototype System

- Strands are regarded as immutable objects, in order to minimize copying associated with editing data on strands, thus editing consists of manipulating <u>pointers</u> to strands
- Garbage collection algorithm may be used to reclaim unused space resulting from deallocation of pointers
- Significant Functions of Prototype include
  - RECORD [media] → [requestID, mmRopeID]
  - PLAY [mmRopeID, interval, media] → requestID
  - STOP [requestID]
- Editing Operations include INSERT, REPLACE, SUBSTRING, CONCATE, DELETE, and REPLACE

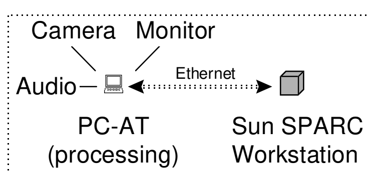
Implementation of Prototype System - Maintenance of Scattering While Editing

- Editing multimedia data can result in discontinuities due to variances in scattering parameter (separation between successive blocks of storage)
- This separation may be closed by copying an adequate number of strands into this interval space

Talk By Jeff Waldrop - For CS 8113D Multimedia

Implementation of Prototype System - System Requirements

- Hardware Consisted of
  - Sun SPARC station
  - PC-AT
  - Video Camera
  - TV monitor



- Software Architecture consists of
  - Multimedia Storage Manager
  - Multimedia Rope Server

Talk By Jeff Waldrop - For CS 8113D Multimedia

Conclusions

This paper

- analyzed specific requirements of multimedia file systems
- defined abstractions for conceptualizing this storage and the requisite parameters
- presented a prototype system that allows exploration of the concepts presented; this prototype is being enhanced to allow network access and access via an interface which allows on-line conversations and conferences between users