Open Hypermedia File System

Tom DeDonno 1/8/96
UCSD/ECE-0408
9500 Gilman Drive 220
La Jolla, Ca 92093
(619)433-4170
tdedonno@sdcc3.ucsd.edu

Abstract

This paper outlines the objectives of an open hypermedia file system. A hypermedia file system addresses the objectives associated with the development of a multimedia file system and takes a step further by providing additional capabilities to meet the needs of future hypermedia systems. Standard multimedia file systems have three objectives:

* accommodating large data objects

* meeting continuous interactive playback requirements

* and synchronization of multimedia data

A hypermedia file system should meet both these objectives and addresses other issues arising in hypermedia. More specifically, it should address: the ability to automatically access and index multimedia data, effective handling of dangling links, Quality of Service negotiations over networks, spatial indexing of temporal objects thereby permitting precise random access of temporal data, open hypermedia, creation of generic linking of non textual objects through file annotation, and robust network security providing multimedia object access lists based on network domains.

Multimedia File System

Multimedia file systems are concerned with handling of large contiguous clusters, continuous time constraints and synchronization of heterogeneous media. When dealing with temporal based data (i.e., video & audio) in a multimedia system the file system must send information to the user at a continuous pace. To enable a viewer to watch video at the standard thirty frames per second, the multimedia file system must access and send thirty frames every second to the display. Multimedia file systems have the option of delaying the start of the video. However, once the video has commenced, jitter (variance in time between sequential delivery) must be less than the playback rate of the medium. [2]

Many forms of multimedia data have very high disk space requirements. Table 1 illustrates the disk space requirements for the four MPEG (Moving Picture Expert Group) standards. MPEG stands to become the most popular form of compression for video and audio streams. Traditional file systems hardly ever deal with dynamic data files larger than 1MByte. However, table 1 illustrates that a 90 minute TV quality movie compressed using MPEG-1 has a disk space requirement of 1GByte. The MPEG-3 standard supporting HDTV (High Definition TV) would require 225MBytes of disk space for just a two minute video presentation. Table 2 lists the evolving hardware capacities of server and local PC disks. These tables illustrate that the storage and playback of movie length high quality MPEG videos are unlikely in the near future. A standard 2GByte PC disk in the year 1998, would not fit even a single MPEG-2 hour and half VCR quality movie. However, interpolating from tables 1 and 2, a multimedia system in the year 1998 with a 2Gbyte disk could hold the following multimedia video streams:

* Short HDTV quality MPEG-3 video segments

* Full length MPEG-4 video conferences

* MPEG-1 movies stored predominantly on CD-ROM with hard disk cooperation

Table1: MPEG Storage Requirements

Standard	Quality	Compress Ratios	Duration		Playback	
			2-Minutes	1.5Hour	Block/sec	msec/block
MPEG-1	TV	1.5Mbps	22.5MB	1GB	192	5.21
MPEG-2	VCR	5Mbps	75MB	3GB	640	1.56
MPEG-3	HDTV	15Mbps	225MB	10GB	1920	.52
MPEG-4	Teleconf	25kbps	375KB	16MB	3	333.3

Another key concern of multimedia file systems is the synchronization of related but disparate data. A multimedia file system could create heterogeneous blocks that contain a mixture of all the objects needed to be synchronized. For example, a text stream will have the corresponding audio built into the text based document structure. The system could also store each type of multimedia data in the appropriate format and include additional information in separate files to establish synchronization. The first method is impractical since all forms of multimedia data are being driven in the direction of independent compression formats (i.e., JPEG, MPEG, Text, RTF, etc.). In addition, the second method places the burden of synchronization directly on the file system.

Table2: Evolving Disk Capacities		
Year	PC($250)	Server($2K)
Jan/95	500MB	2GB
1998	2GB	8GB
2001	8GB	16GB

Hypermedia File Systems

The prime objective of a hypermedia file system is to simplify and enhance the performance of a hypermedia system by providing libraries and mechanisms to do the key tasks. A hypermedia system is concerned with combining various forms of media through interlinking yielding an enhanced platform for presentation and dispersion of information. To accomplish this goal a hypermedia file system must inherit and enhance many attributes of the previously discussed multimedia file system. Besides, enhancing a multimedia file systems, a hypermedia file system is devoted to accomplishing two other objectives: Provide a vehicle to aid in the creation of active hypermedia systems. Solve the special needs of future hypermedia systems that will gradually replace the current hypertext (i.e., Mosaic, Netscape and Gopher) systems.

Enhancement of Multimedia File System

As previously, stated multimedia file systems are concerned with: allocation of long adjacent blocks, synchronization of heterogeneous media, and meeting continuous time constraints. To effectively meet the MPEG playback rates of table 1, three basic requirements must be satisfied: specialized graphic hardware, adequate storage transfer rates, minimization of disk access times.

Specialized Graphic Hardware

Future graphic hardware will easily playback MPEG at thirty frames per second. Already many vendors are selling high performance PC graphic cards with thirty frames per second MPEG decoders selling for under $300. For example, the Stealth64 Video 3200 is priced at $299.

Adequate Transfer Rates

The transfer rate of a 1.2GByte IDE disk is 12MBytes/sec (i.e., 12KBlocks/sec) these transfer rate can adequately support all classes of MPEG except the high quality MPEG-3.

Minimization of Disk Access

Unfortunately, disk access times are substantially slower than transfer rates. Permanent storage systems must employ some form of nearby block allocation to guarantee continuous playback. With defragmentation software, computer systems can meet interactive continuous time constraints and still maintain the basic file system structure. A key ingredient for the success of any new idea or system is to maintain compatibility with existing standards and provide enhancements to accommodate the needs of evolving systems. In addition, to maintaining standards, the designer must consider future hardware directions and capabilities. Integrated device electronics is increasing at overwhelming rates. However, mechanical devices are improving very slowly. The main permanent storage device of almost all computers is the hard disk. Hard disks are plagued with two mechanical shortcomings: seek time, disk head movement between concentric tracks; and latency delay, the time it takes for the constant rotating disk platters to rotate to the correct segment of a track. Because of these two mechanical restrictions, disk performance has not kept pace with evolving CPU performance. Disk performance has been doubling every five and half years, with CPU performance doubling almost every eighteen months. Without buffering a system must access, read and transfer each block within the time constraints listed in the last column of table 1. For MPEG-1, this value is 5.21 msec/block. On a modern day disk the time to read, access and transfer one random block at a time is approximately 17.64msec. This value is roughly three times greater than the maximum allowed delay. Nota bene, even if one eliminates the seek time, the corresponding latency delay is still too slow for continuous playback.

Since current disks cannot meet MPEG continuous playback by reading one block at a time will future disk provide the necessary performance? By extrapolating the performance increases between the years 1988 and 1995. A 1999 future disk will read, access and transfer a random block at 9.025msec (i.e., .015+4.4+4.61). Even this future disk would not meet the required MPEG-1 playback rate of 5.21msec. Obviously in order for interactive video to become a reality on future computers the file system must minimize seek times and latency delays. Since disk rotation is purely mechanical latency delays will become the weakest link. The best way to increase performance in anything is to remove the weakest link. In hard disks, the weakest link will eventually become the latency delay. Latency delays can almost be eliminated by positioning all strands (i.e., video segment lasting approximately one second) onto only one cylinder. The blocks would be in ascending order so the entire strand could be read during one rotation. The length of the strand would be sufficiently large enough so that latency delays don't exceed the required playback rate of the device.

Active Hypermedia

Most hypertext systems are passive. An author must manually add links to the system in order for the user to access information. In an active hypermedia system, the linking of information is automatic. A hypermedia file system can aid in the creation of active hypermedia systems by providing five basic features: First, automatic indexing of key hypertext (i.e., HTML) files and homepages. Second, direct communication links with robots, wanders, and spiders. Robots, wanders and spiders are often called web crawlers. These crawlers search computers for key HTML files and send their findings back to a home system. The hypermedia file system can provide key information directly to these crawlers alleviating the requirement that the crawlers physically access each host machine. Third, additional file mappings based on file types not only directory structure. When one is using a MPEG viewer, a file structure showing all nearby MPEG videos is superior to the standard tree structure. Fourth, non textual hypermedia files such as MPEG, JPEG, and GIF cannot be accessed through standard text based queries. Database management systems allow query access to text based system through text based queries. To allow text based query access of non textual data, the hypermedia file system will maintain a corresponding keyword and annotated text file associated with each non textual multimedia object. A query will consist of accessing the annotated files and a selection will result in sending the associated multimedia object. Fifth, current hypertext systems contain only specific links. A specific link is created by an author at a specific location within a specific file pointing to a specific document. Active systems must allow the inclusion of generic and dynamic linking. A generic link allows the author to attach a link to a single instance of an object that is then automatically attached to all instances of the object.[1]

Evolving Hypermedia Systems

Besides satisfying the need of active hypermedia, future hypermedia systems must seek to provide the following capabilities: resolution of dangling links, spatial indexing (i.e., random disk access) of temporal objects, quality of service (QOS) negotiations, enhanced communication with hypertext protocols, and network based security access lists.

Dangling Links

A key problem with standard hypertext systems is dangling links. Almost all HTML files have pointers to other documents. Significant research has already been done in these area. A hypermedia file system must inherit some form of link bases to eliminate dangling pointers.

Indexing of Temporal Objects

Traditional file systems provide random and sequential access of spatial/sequential based data. Skipping ahead one or two pages in a spatial/sequential document is relatively easy on a spatial based medium. However, skipping ahead five minutes into a video is difficult. The video is a temporal based medium and random access to disk are spatial. A system can provide random access to temporal data in one of two methods: First, it can estimate the starting location of a specific frame by considering average compression rates. For example, MPEG-1 compresses at 1.5Mbps, a random access point at 3Mbits would be roughly two seconds into the video. The major drawback of this approach is the wide variance between compressed files. Going from an average compression ratio to a spatial file location is only an estimate. The system could very well deliver video that is nowhere near the desired time frame. In addition, what part of the corresponding audio stream must be sent to maintain synchronization. Second, the hypermedia file system creates an index of a MPEG video similar to a standard IBM VSAM. Each intracoded frame would have an index file entry consisting of a doublet containing file offset and frame number. Both these values are accurately obtained. The physical spatial disk location is a standard file system component. The individual times of each MPEG intracoded frame is obtained by simply keeping track of all frames from the start of the video. MPEG compression consists of intercoded and intracoded frames. An intercoded frame relies on information on previous and/or future intracoded frames to reconstruct itself. An intracoded frame is encoded and decoded with information pertaining to only itself. MPEG viewers can start playback at any MPEG intracoded frame. This method would also lend itself well to fast forward and reverse. In fast forward mode the system can send a specific sequence of Intracoded frames depending on the speed of fast forwarding. Note this method would provide the intrinsic capability of watching an entire video at a fast forward rate even if the corresponding physical hardware was unable to meet this continuous demand. For casual use, the size of the index file can be greatly reduced by generating a ten-second index file. For every three hundred frames HFS records the file offset of the next intracoded frame. Typical viewers would not require a fast access mechanism whose precision exceed ten seconds.

Open Hypermedia

Almost all hypertext and hypermedia systems developed are closed hypermedia systems. A closed hypermedia system requires that the original document be marked up with linking information. In an open hypermedia system such as Microcosm, the link information is maintained in a linkbase. A linkbase is a link database file that maintains information on all links. [1] Future hypermedia systems will need to access and organize entire libraries of information. These libraries of information will prohibit the markup of the originals. The originals may be on read only material or the system would require the author's permission to markup a document. A solution to this problem resides within the hypermedia file system's ability to create independent link bases that perform hyper linking on read only material. When appropriate the hypermedia file system would transparently combine the link bases and read only material.

Quality of Service (QOS) negotiations

A major drawback of the world wide web (WWW) is the slow delivery of images. Users accessing the WWW at non prime hours with a direct digital connection approach interactive access. Users accessing the WWW through the standard analog phone lines are usually burdened with slow image delivery. Unfortunately, most WEB users use slow analog phone connections. Two popular HTML hypertext engines Mosaic and Netscape provide a user based long term scheduler. The Mosaic and/or Netscape screen will display text at an interactive pace, however, images are initially displayed as just an image bar. The actual image is not displayed unless the user physically waits for its arrival. With the coming arrival of video information to the WWW, even users with direct digital connections will wait several minutes for video delivery. Even today, expensive teleconferencing systems are burdened with delays on video delivery. If optical networks (i.e., ATMs) are installed and deliver the promised network bandwidth, Internet delivery of real time video and images may become a reality. However, without optical networks a suitable compromise besides a user based long term scheduler, is to setup a quality of service (QOS) negotiation between the hypermedia file system and the requesting hypermedia system engine. For example, the JPEG standard can compress images at various resolution. On playback the hypermedia engine could request that all images be sent at only 25% resolution. As another example, when the network cannot support video at thirty frames/second, the hypermedia engine and file system can negotiate quality of service of fifteen frames/second.

Network Access Lists

Today computer systems deal with security at predominantly the user level. Computers don't sufficiently handle network security. Network security deals predominantly with allowing certain systems to access only part of the file system. Future hypermedia systems will require a more robust network security system. The Unix operating system associates an access list with each file. An access list consists of a three level user domain (i.e., everyone, group and user) and three file access rights (i.e., read, write and executable). An access list associates a user (general term domain) and access rights to each object. A capability list associates object and access rights to each domain (specific term user). Future hypermedia file systems must extend this concept to a networked environment. The domain associated with a file is extended into a network entity consisting of: network domain (i.e., com, edu), computer network (i.e., UCSD, sun, etc.) and specific network node (i.e., CSE, SDCC3). The access rights would be identical or similar to a standard Unix file system.

Conclusion

Future file systems must position themselves as middle ware applications that lie on top of the operating system kernel. These middle ware applications (e.g., Hypermedia File System) can provide the necessary mechanisms to meet the needs of future application programs and still maintain compatibility with existing computer systems. This middle ware hypermedia file system can meet the needs of multimedia storage and retrieval by providing a cylinder approached method of storing and retrieving video segments. An optimal number of blocks are restricted to exactly one cylinder. These blocks all occur within the same time frame of a video and have a performance limitation of only one latency delay. As these blocks are read off of one cylinder they are sorted and sent to pinned memory buffers associated with the video. Besides multimedia storage and retrieval, these systems must also provide an active hypermedia system foundation, that can automatically integrate new information into existing hypermedia databases. The hypermedia file system can provide active hypermedia foundations by providing: Annotated key files of non textual based objects and generic linking of non textual objects. The final responsibility of a hypermedia file system is to meet the specialized requirements of future comprehensive hypermedia systems. These specialized requirements include: the effective handling of dangling links, spatial based indexing of temporal objects, open hypermedia facilitates, quality of service negotiations, easily accessible network communication, and network based access lists.

References

[1] G.A. Hutchings, A.J.D. Warren & S.T. Rake. Microcosm User Guide Release 3.0 for Windows, University of Southampton, MLS Technology Ltd. May 1994.

[2] Harrick M. Vin and P. Venkat Rangan. Designing a Multi-User HDTV Storage Server. IEEE Journal on Selected Areas in Communications, 11(1):153-164, January 1993.

[3] Jeffrey R. Bach, Santanu Paul, and Ramesh Jain, A Visual Information Management System for the Interactive Retrieval of Faces, IEEE Transactions of Knowledge and Data Engineering Vol. 5, No. 4, Aug. 1993, pp 619-628

[4] Ramesh Jain Editor, NSF Workshop on Visual Information Management Systems, Feb. 24-25 1992, Redwood Ca.