Multimedia System Architecture

Harrick M. Vin

Distributed Multimedia Computing Laboratory
Department of Computer Sciences
University of Texas at Austin
Taylor Hall 2.124
Austin, Texas 78712-1188

E-mail: vin@cs.utexas.edu, Telephone: (512) 471-9732, Fax: (512) 471-8885

Citation: *In Proceedings of the SPIE International Symposium on Photonics for Industrial Applications, Critical Review on Defining Global Information Infrastructure: Infrastructure, Systems, and Services, Boston, November 1994*

This research was supported in part by the National Science Foundation (Research Initiation Award CCR-9409666), NASA, Mitsubishi Electric Research Laboratories (MERL), Sun Microsystems Inc., and the University of Texas at Austin.

Multimedia System Architecture

Harrick M. Vin

Department of Computer Sciences, The University of Texas at Austin
Taylor Hall 2.124, Austin, Texas 78712-1188
E-mail: vin@cs.utexas.edu, Phone: (512) 471-9732, Fax: (512) 471-8885

ABSTRACT

In this paper, we describe a multimedia system architecture consisting of: (1) an *information management subsystem*, (2) a *storage subsystem*, and (3) a *network subsystem*. Whereas the information management subsystem provides means for identifying the set of multimedia objects that may be pertinent to a client's query, the storage subsystem ensures that multimedia objects are efficiently stored and retrieved from secondary storage devices. The network subsystem, on the other hand, guarantees timely delivery of the multimedia objects accessed by the storage subsystem to each of the client sites. The main goal of this paper is to identify and discuss the research issues involved in designing each of these three subsystems.

Keywords: multimedia server, high speed networks, transport protocols, multimedia databases, information management, compression techniques

1 INTRODUCTION

Since the debut of the first usable computer almost half a century ago, the world has witnessed dramatic improvements in computer and communications technologies. Whereas breakthroughs in computer technology have stimulated the integration of digital continuous media (such as audio and video) with computing, rapid advances in communication technology have made available high-bandwidth, fiber-optic networks at modest cost.[21] The synergy between the advances in computer and communications technologies promises to lay a foundation for designing sophisticated multimedia information management systems (or *multimedia systems*, for short) in a wide range of application domains. For instance:

- Designing a large-scale multimedia repository of the available lectures, videos, interactive classes, etc., and providing powerful means for navigating and browsing through the information space will significantly aid education. In fact, we envision that entire libraries (such as the Library of Congress) and archives will be stored on distributed multimedia servers, access to which will be provided as a network service over national high-bandwidth networks (e.g., the NSF/ARPA/NASA Digital Library Initiative). This would enable schools, universities, and industrial organizations to share a wide range of educational and learning material.

- Multimedia systems will provide powerful means for supporting computational prototyping and scientific visualization (e.g., of large-scale physics and chemistry simulations), and hence, will tremendously increase the productivity of engineers and scientists. Similarly, in the fields of astronomy and environmental sciences, efficient techniques for storage and retrieval of complex multimedia objects will assist scientists in analyzing images recorded from satellites (each of which may contain several tens of megabytes of information).
- Modern medicine extensively employs imaging methods (e.g., X-rays, CT Scans, etc.) in the diagnosis process. A multimedia system will significantly assist medicine practitioners in the process of diagnosis by providing efficient mechanisms to access a variety of information about patients. For instance, a multimedia system optimized for medical applications will support queries such as: where was the tumor initially located (i.e., detection), what happened to the tumor (i.e., temporal analysis), assessment of the tumor (i.e., benign or malignant), and the assessment of the therapeutic response (i.e., did the therapy reduce the tumor).
- Commercial services such as trade services, advertising, product announcements, customer support, and travel reservations will all be made available through multimedia systems. To illustrate, vendors will advertise items such as appliances, automobiles, homes, etc., by listing them in an on-line digital multimedia catalog. Potential buyers will browse through the catalog, selectively view the accompanying full-motion video demonstration, and tailor the demonstration to the desired level of detail. In entertainment, digital multimedia services promise to offer greater selectivity to clients by providing access to a wider variety of entertainment programs. In addition to providing fine-grained control (such as pause, resume, fast-forward, etc.) over the program being viewed, these advances will also inspire the development of an entirely different spectrum of sophisticated *personalized* services, in which clients will be able to customize the service by choosing from amongst a multitude of offerings and only procuring what they desire and no more. Two of the most prominent illustrations of such sophisticated information services are personalized newspapers and video channels.

In this paper, we describe an architecture for designing such large-scale multimedia systems of the future. Our proposed architecture consists of three main components: (1) an *information management subsystem*, (2) a *storage subsystem*, and (3) a *network subsystem*. Whereas the information management subsystem provides means for identifying the set of multimedia objects that may be pertinent to a client's query, the storage subsystem ensures that multimedia objects are efficiently stored and retrieved from secondary storage devices. The network subsystem, on the other hand, guarantees timely delivery of the multimedia objects accessed by the storage subsystem to each of the client sites. In what follows, we discuss critical design issues for each of these three subsystems.

2 INFORMATION MANAGEMENT SUBSYSTEM

To manage the information explosion yielded by the advances in storage and communication technologies, future information management systems will be required to support powerful means of navigating through the information space. The main function of such systems will be to identify specific information objects that may satisfy the client's requirements. However, due to the enormity of data, it may be computationally infeasible to examine each data item within the information repository in detail. Consequently, while all of the data is precious, only a fraction of it may be computed against or examined at any particular moment. Furthermore, the inherent nature of multimedia data may require it to be analyzed in several dimensions (e.g., spatial, temporal, etc.) as well as at various resolutions. To address these requirements, future information management systems will be required to provide systematic support for managing data at various levels of detail. We refer to systems that provide such support *multiresolution* or *MR* systems.[18]

Multiresolution refers to the notion of viewing data at different levels of information content. The concept of multiresolution systems is founded on the assumption that it is more efficient to retrieve and compute against low-resolution data as compared to their high-resolution counterparts. Specifically, it is assumed that if X is a low resolution version of Y, then X requires less storage space as compared to Y. Furthermore, accessing as well as processing X requires significantly smaller computational resources (compute power, network bandwidth, etc.) as compared to Y. Multiresolution systems exploit this general relationship between multiple resolution levels of data and improve performance by computing against lower-resolution data whenever possible.

The meaning of data, and hence the notion of resolution, however, is always application dependent. For example, in the case of a histogram, the number of bins determine its resolution. When there are few bins and many data objects are lumped together, a histogram denotes a low resolution representation of a set. On the other hand, when there are a large number of bins, a histogram provides a great deal of useful information about the set, and hence denotes a higher resolution. At the highest possible resolution level, each bin contains exactly one object, thereby completely defining the membership of a set. Similarly, images can be encoded at various resolution levels, for instance in chroma (i.e., number of bits/pixel) or spatial (i.e., number of pixels/image) dimensions. In general, a multiresolution data type can be viewed as a partially ordered set of full-resolution (i.e., *total*) and lower-resolution (i.e., *partial*) data elements describing some domain at varying levels of detail.[18,19]

A key advantage of defining such a multiresolution data type is that it admits the concept of *progressive refinement* that is not available in the conventional data models. In most conventional systems, a query has a single, well-defined, *result*. However, in the multiresolution model, queries may have many lower-resolution approximations, referred to as *responses*. Having produced a low-resolution response, the process of progressive refinement can then produce a chain of responses, each of which approximates the result of the query and improves upon the last response.

To implement a multiresolution information management system, a designer will be required to define domain-specific multiresolution primitive types, including the relations that specify the ordering of various resolution levels. Additionally, the designer will be required to provide application-specific functions that can produce low-resolution data from high-resolution data. In fact, the practicality of a multiresolution type in a particular application will depend on the compactness of partial data elements and the efficiency of analyzing and extracting information from lower-resolution data objects.

Once a multiresolution data model is defined, the system designer will be required to define query language extensions that permit a client to control the resolution of responses. The relationship between time expense and response resolution may be complex. One potential solution is to extend the notion of "value functions" (that map time expense into value) used in real-time systems.[8,10] Such functions capture typical imaginable "valuations", such as deadlines after which the value is zero, deadlines with rapid drop off in value, value peaking around a particular time, a demand that resolution be total, a gradual increase in value with resolution, specific resolutions at specific times only, etc. Finally, as a result of associating value functions with each query, a query optimizer must, in addition to identifying what data needs to be retrieved, determine the process of its retrieval.

In summary, to efficiently navigate through and access multimedia objects from large repositories, future information management systems will have to support: (1) a multiresolution data model, (2) multiresolution storage structures, (3) query language extensions that allow the specification of resolution/performance tradeoffs, and (4) a framework for query optimization that can successfully meet those specifications. Moreover, the realization of such systems will require the design of storage and transport subsystems which can provide efficient access to multimedia objects, possibly at different resolution levels. Architectures and algorithms for designing such storage and the network subsystems are detailed in the following sections.

3 MULTIMEDIA STORAGE SUBSYSTEM

The fundamental problem in developing a multimedia storage subsystem (also referred to as a *multimedia server*) is that images, video, audio, and other similar forms of data differ from numeric data and text in their characteristics (e.g., format, size, etc.), and hence, require totally different techniques for their organization and management. The most critical of these characteristics is the *continuity requirement*: since digital audio and video streams consist of a sequence of media quanta such as video frames or audio samples, which convey meaning only when presented continuously in time (unlike text in which spatial continuity is sufficient), a multimedia server must ensure that recording and retrieval of media streams to and from disks proceed at their real-time rates. Although semantically different, both of these operations are mathematically equivalent with respect to their real-time performance requirements.[1] Consequently, for the sake of clarity, we will only discuss technical challenges for ensuring continuous retrieval of media information from disk for real-time playback. Analysis for real-time recording can be carried out similarly.

Continuous playback of a media stream consists of a sequence of periodic tasks with deadlines, where tasks correspond to retrievals of media blocks from disk and deadlines correspond to their scheduled playback times. Generally, since the data transfer rates of disks are significantly higher than the requirements of a single stream (e.g., the maximum throughput of modern disks is of the order of 3-4 MBytes/s, while that of an MPEG encoded video stream is 0.5 MBytes/s, and uncompressed CD-quality stereo audio is about 0.2 MBytes/s), employing modest amount of buffering will enable conventional file and operating systems to support continuous storage and retrieval of isolated media streams.

In practice, however, a multimedia server has to process requests from several clients simultaneously. In the best scenario, if all the clients request the retrieval of the same media stream, the server needs only to retrieve the stream once from the disk and then multicast it to all the clients. However, more often than not, different clients may request the retrieval of different streams; and even when the same stream is being requested by multiple clients (such as a popular movie), there may be phase shifts among their requests (e.g., each client viewing a different part of the movie at the same time).

A simple mechanism to guarantee that the real-time requirements of all the clients are met is to dedicate a disk head to each stream. This, however, limits the total number of streams to the number of disk heads. In general, since the data transfer rate of disks are significantly higher than the requirements of a single stream, the number of streams that can be serviced simultaneously can be significantly increased by multiplexing a disk head among several streams. However, given the maximum rate of data transfer from disks as well as the data rate requirement of each stream, a server can service only a limited number of clients simultaneously. Hence, before admitting a new client, a multimedia server must employ admission control algorithms to decide whether a new client can be admitted without violating the continuity requirements of any of the clients already being serviced.[1,4,16,22,26,27]

The precise admission control criteria, however, is dependent on the quality of service requirements of the clients. In general, the types of service provided by a server can be broadly classified into the following four categories:

- *Deterministic:* All deadlines are guaranteed to be met. For this level of service, the admission control algorithm must consider worst-case scenarios before admitting new clients.[26]
- *Statistical:* Deadlines are guaranteed to be met with a certain probability. For example, a client may subscribe to a service that guarantees that 90% of deadlines will be met over an interval. To provide such guarantees, admission control algorithms must consider the stochastic behavior of the system while admitting new clients.[25]
- *Predictive*: Service is provided only if the prediction from the measured server performance characteristics indicate that the service requirements of all the clients can be met satisfactorily.[24] The basis for prediction is generally the average performance characteristics (as opposed to the precise distributions employed by statistical admission control algorithms).

- *Best Effort:* No guarantees are provided for meeting deadlines. The server just "tries its best" - i.e. it schedules such accesses only when there is time left over after servicing all guaranteed, statistical, and predictive clients.

Regardless of the type of service, once a client is admitted by the server, media information retrieved from disk for the client is stored in intermediate buffers prior to transmission. Ensuring timely delivery of the media information from these intermediate buffer to client sites is the function of the network subsystem. In what follows, we outline techniques for delivering multimedia object over low as well as high bandwidth networks.

4 NETWORK SUBSYSTEM

4.1 Delivering multimedia objects over low bandwidth networks

On low bandwidth networks, in the simplest case, a multimedia object can be presented to clients by transmitting and buffering the entire object at client sites before initiating the presentation. Although relatively straightforward to implement, such a scheme may limit the size of the object being delivered to the buffer space availability at the client sites. Furthermore, since the buffering process itself may be highly time consuming, such an approach may yield unacceptable initiation latencies.[20]

Consequently, the problem of efficiently delivering a multimedia object to client sites becomes one of preventing starvation while at the same time minimizing buffer space requirement and initiation latency. Clearly, if the bandwidth available on the network is larger than the peak data rate requirement of a multimedia object, the presentation can be initiated without any buffering. However, if the dynamic variation in the playback rate requirements, resulting from the structure of the multimedia object, occasionally require higher data transfer rate than the available bandwidth, then the network subsystem may be required to employ *time shifting* policies. Such policies utilize periods of low channel utilization to deliver media information prior to their presentation times. The main goal of such techniques is to derive transmission schedules which ensure jitter free presentation of complex multimedia objects at client sites.

4.2 Managing multimedia communication over high speed networks

Efficient utilization of high speed network resources, on the other hand, require the network to statistically multiplex several media communication channels. However, while doing so, the network must ensure that the quality of service guarantees (in terms of bandwidth, end-to-end delay, media loss, and asynchrony) provided to clients are not violated. Whereas the minimum guaranteed bandwidth must be large enough to accommodate media streams at acceptable resolutions, the end-to-end delay must be small enough for interactive communication. In order to avoid breaks in continuity of video display, media unit loss must be sufficiently small. Simultaneous display of multiple media streams also requires that the asynchrony between their display be bounded within human perception tolerances.[15,17]

Among these Quality of Service (QoS) requirements, bandwidth, delay and loss guarantees can be regarded as fundamental, since they have to be provided by the network layer.[3] However, the burstiness in multimedia traffic, yielded by variable bit rate compression techniques (e.g., JPEG, MPEG, etc.), complicates the development of techniques for providing such guarantees.[3,6,7,9,12,14] One possible approach for simplifying the design of techniques for providing QoS guarantees is to smooth the multimedia traffic characteristics by buffering media information prior to transmission. Although such an approach may yield higher end-to-end delay (and hence, may not be appropriate for interactive multimedia applications, such as video conferencing),[13] it is likely to be very promising for retrieving information from large-scale storage servers. In fact, we conjecture that, as a result of smoothing, the transmission requirements of media streams encoded using variable bit rate compression schemes will be captured by a finite set of data transfer rates. Such a traffic characterization will simplify the network admission control as well as enhance the statistical multiplexing gain of finite-buffer packet switches.

4.3 Encoding techniques and error control

An important component of transport protocols for multimedia communication over low as well as high bandwidth networks is *error control*. Most conventional transport protocols are founded on the presumption that all the client traffic must be delivered without any loss, and hence, employ Automatic Repeat reQuest (ARQ) techniques for ensuring reliable communication. ARQ is a closed-loop technique requiring a feedback channel from the receiver to the sender. Using this technique, client traffic is sent with a small amount of redundant information for error detection at the receiver, and the sender is required to retransmit the client traffic when needed. Notice, however, that since multimedia information generally contains a large amount of redundancy, a multimedia transfer protocol can, relatively easily, recover from transmission errors (e.g., dropped or corrupted packets) introduced by the network. Furthermore, the stringent real-time requirements for multimedia communication render ARQ techniques highly ineffective.

To design retransmission-free transport protocols for digital multimedia, several research projects have begun investigating image encoding techniques that enable client sites to reconstruct the image sequence even when a fraction of the transmitted information is lost or corrupted. Most of these protocols can be broadly classified as employing either *explicit* or *implicit* forward error correction (FEC) techniques. In the case of explicit FEC, redundant information (e.g., parity bits) is transmitted with the original traffic such that the images can be reconstructed at the receiver even if some of the traffic is lost or corrupted by errors.[2] However, for explicit FEC to be effective, the amount of redundant information that can be transmitted with the original data must be very small. This is because the additional data traffic yielded by the redundant FEC information increases the overall load, which in turn may worsen the loss rate. Furthermore, the usefulness of FEC diminishes when losses are highly correlated since more redundant data must be sent to correct for the loss.

Implicit FEC, on the other hand, is provided by *layered encoding* and *image scrambling* techniques. Layered encoding is founded on the notion that the output of a source coder can be partitioned into an essential layer (which is critical for conveying basic information content of the image) and enhancement layers (which when added to the the essential layer recreate the signal more fully).[5,11,28] Consequently, transport protocols that employ layered encoding techniques arrange the contents of the essential and enhancement layers into packets of high and low priority, respectively, and then ensure that all the high priority packets are timely delivered to each of the client sites. Since the information contained in high priority packets is sufficient to create a reasonable approximation of the image at the client sites, such a scheme enables the protocol to tolerate isolated loss of packets containing low priority information. On the other hand, minimizing the perceptible artifacts yielded by bursty losses will require images to be scrambled such that neighboring pixels are transmitted in separate packets spaced out in time.[23] These techniques dramatically reduce the likelihood of neighboring pixels being lost in a single burst, and thereby make it possible to reconstruct the lost image data by extrapolating from neighboring pixels.

In summary, to ensure timely delivery of media information to client sites, network subsystem will be required to employ transport protocols that are optimized for multimedia communication over low as well as high bandwidth networks. These protocols will integrate encoding techniques with the service guarantees provided by a network for packets or ATM cells; and, in turn, will provide end-to-end service guarantees to each client.

5 CONCLUDING REMARKS

Designing multimedia information management systems of the future poses a variety of inter-disciplinary research challenges. Attaining a breakthrough in designing and evaluating such systems will require orchestrating a coherent and comprehensive effort that integrates expertise in media compression, multimedia data management, parallel storage and I/O architectures, network protocols, and systems engineering. Design and implementation of such a multimedia information management system is the focus of our research at the UT Austin Distributed Multimedia Computing Laboratory.

6 REFERENCES

[1] D. Anderson, Y. Osawa, and R. Govindan. A File System for Continuous Media. *ACM Transactions on Computer Systems*, 10(4):311–337, November 1992.

[2] E.W. Biersack. Performance Evaluation of Forward Error Correction in ATM Networks. In *Proceedings of ACM SIGCOMM'92, Computer Communications Review, Vol. 22, No. 4*, pages 248–257, August 1992.

[3] D. Ferrari and D. C. Verma. A Scheme for Real-Time Channel Establishment in Wide-Area Networks. *IEEE Journal on Selected Areas in Communications*,

8(3):368–379, April 1990.

[4] J. Gemmell and S. Christodoulakis. Principles of Delay Sensitive Multimedia Data Storage and Retrieval. *ACM Transactions on Information Systems*, 10(1):51–90, 1992.

[5] M. Ghanbari. Two-Layer Coding of Video Signals for VBR Networks. *IEEE Journal on Selected Areas in Communications*, 7(5):771–781, June 1989.

[6] S.J. Golestani. A Framing Strategy for Congestion Management. *IEEE Journal on Selected Areas in Communications*, pages 1064–1077, September 1991.

[7] R. Guerin, H. Ahmadi, and M. Naghshineh. Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks. *IEEE Journal on Selected Areas in Communication*, pages 968–981, September 1991.

[8] Jiandong Huang, John A Stankovic, Don Towsley, and Krithi Ramaritam. Experimental evaluation of real-time transaction processing. In *Proceedings of the Real-Time Systems Symposium, Santa Monica, CA*, pages 144–153, December 1989.

[9] J. M. Hyman, A. A. Lazar, and G. Pacifici. Joint Scheduling and Admission Control for ATS-based Switching Nodes. In *Proceedings of ACM SIGCOMM*, pages 223–234, August 1992.

[10] E. Douglas Jensen, C. Douglass Locke, and Hideyuke Tokuda. A time-driven scheduling model for real-time operating systems. In *Proceedings of the Real-Time Systems Symposium*, pages 112–122, 1985.

[11] T. Kinoshita, T. Nakahashi, and M. Maruyama. Variable Bit-rate HDTV CODEC with ATM Cell Loss Compensation. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(3):230–236, June 1993.

[12] J.F. Kurose. On Computing Per-session Performance Bounds in High-Speed Multi-hop Computer Networks. *In Proceedings of SIGMETRICS'92, Performance Evaluation Review*, pages 128–139, June 1992.

[13] S.S. Lam, S. Chow, and D.K.Y. Yau. An Algorithm for Lossless Smoothing of MPEG Video. In *To Appear in the Proceedings of ACM SIGCOMM'92, London*, 1994.

[14] A. Parekh and R. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks – The Single Node Case. In *Proceedings of IEEE INFOCOM '92*, pages 915–924, May 1992.

[15] C. Partridge. Isochronous Applications Do Not Require Jitter-Controlled Networks. *Request for Comments, RFC 1257*, September 1991.

[16] P. Venkat Rangan and Harrick M. Vin. Designing File Systems for Digital Video and Audio. In *Proceedings of the 13th Symposium on Operating Systems Principles (SOSP'91), Operating Systems Review, Vol. 25, No. 5*, pages 81–94, October 1991.

[17] P. Venkat Rangan, Harrick M. Vin, and Srinivas Ramanathan. Designing an On-Demand Multimedia Service. *IEEE Communications Magazine*, 30(7):56–65, July 1992.

[18] Robert L. Read, Donald S. Fussell, and Avi Silberschatz. A multi-resolution relational data model. In *Proceedings of the 18th International Conference on Very Large Data Bases, Vancouver, Canada*, pages 139–150, August 1992.

[19] Robert L. Read, Donald S. Fussell, and Avi Silberschatz. The fact set approach to the sandbag imprecise set representation and its complexity. 1993. In preparation.

[20] J. Rosenberg, G. Cruz, and T. Judd. Presenting Multimedia Documents Over a Digital Network. *Computer Communications*, 15(6):374–380, July/August 1992.

[21] D.R. Spears. Broadband ISDN - Service Visions and Technological Realities. *Journal of Digital and Analog Cabled Systems*, 1988.

[22] F.A. Tobagi, J. Pang, R. Baird, and M. Gang. Streaming RAID: A Disk Storage System for Video and Audio Files. In *Proceedings of ACM Multimedia'93, Anaheim, CA*, pages 393–400, August 1993.

[23] C.J. Turner and L.L. Peterson. Image Transfer: An End to End Design. In *Proceedings of ACM SIGCOMM'92, Baltimore*, pages 258–268, August 1992.

[24] Harrick M. Vin, Alok Goyal, Anshuman Goyal, and Pawan Goyal. An Observation-Based Admission Control Algorithm for Multimedia Servers. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS'94), Boston*, pages 234–243, May 1994.

[25] Harrick M. Vin, Pawan Goyal, Alok Goyal, and Anshuman Goyal. A Statistical Admission Control Algorithm for Multimedia Servers. In *Proceedings of the ACM Multimedia'94, San Francisco*, October 1994.

[26] 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.

[27] P. Yu, M.S. Chen, and D.D. Kandlur. Design and Analysis of a Grouped Sweeping Scheme for Multimedia Storage Management. *Proceedings of Third International Workshop on Network and Operating System Support for Digital Audio and Video, San Diego*, pages 38–49, November 1992.

[28] Q. Zhu, Y. Wang, and L. Shaw. Coding and Cell Loss Recovery in DCT-Based Packet Video. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(3):248–258, June 1993.