Lightweight Video Service for Multi-Media Digital Libraries*

Shitij Mutreja, Stuart Bailey, Robert Grossman†, and Dave Hanley

1995

Abstract

In this paper we present the design, architecture, implementation and performance of a digital library of video data. Our goal is to provide a lightweight video service for multimedia digital libraries built upon an ATM infrastructure, providing nonlinear editing and viewing of video. To accomplish this objective we have used a low overhead, high performance persistent object manager to manage the underlying video data.

To test these ideas, we used the lightweight persistent object manager PTool, developed by the Laboratory for Advanced Computing at the University of Illinois at Chicago, to create persistent object stores of video data on a cluster of Unix workstations connected with an ATM switch. To improve performance, we striped the video data across the cluster.

We verified that we were able to manage gigabytes of video data without performance degradation and that striping improved performance linearly up to the bandwidth of the equipment. We also compared the performance of ATM networked and ethernet networked clusters.

Finally, we mention that viewing the video data as a persistent collection of objects (i.e. each frame of video as an object) provides attribute based retrieval of data, and hence allows for very simple editing, querying and retrieval of video data by attribute. It allows for the use of object associations to combine video, audio, text, etc.

1 Introduction

In this paper we present the goals, design, implementation, and experimental results of a lightweight video service for work group oriented, multi-media digital libraries. Using modular and layered lightweight services is an emerging trend in building distributed, scalable software. We have developed a modular and

*This research was supported in part by NASA grant NAG2-513, DOE grant DE-FG02-92ER25133, and NSF grants IRI 9224605 and CDA-9303433, IBM, Oak Park Research, Inc., the Laboratory for Advanced Computing at UIC, and by the National Scalable Cluster Project.

†R. Grossman is also with Oak Park Research, Inc.

layered architecture for distributed digital libraries, which is illustrated in Figure 1. Rather than build a specialized application server providing distributed video delivery, our interest was in understanding whether for certain purposes video could be provided as a lightweight service within this architecture.

There has been attention paid to video servers designed to give large numbers of distributed users simultaneous access to centrally located commercially produced video data. On the other hand, our interest was in a system designed for collaborative activities, computer-aided education, and other work group oriented applications. In these types of applications, the video data itself may be distributed, and delivery of the video data to small groups of distributed users is typical.

Two technologies were fundamental to our approach:

- A high-bandwidth Asynchronous Transfer Mode (ATM) networking infrastructure provides a straightforward solution to the bandwidth and constant rate delivery requirements that are necessary for video serving. No specialized software is required as would be the case for a shared medium network.

- Lightweight data management using persistent object stores is a simple way to manage the large amounts of data required to provide video service to the desktop. At this same time, attribute based data access, which is a standard service in our digital library, provides a simple means of supporting nonlinear editing of video data.

To summarize, lightweight data management using persistent object stores and a high-bandwidth Asynchronous Transfer Mode (ATM) networking infrastructure provide simple, effective layers on which to build lightweight video service for multi-media digital libraries.

In this paper, we describe our goals in Section 2, related work in Section 3 and our design in Section 4. Section 5 describes our implementation and Section 6 the experimental results.

2 Goals and Requirements

As part of the National Scalable Cluster Project, we are continuing the development of several digital libraries, including scientific digital libraries for high energy physics data [?], aeronautics data and computational fluid dynamics data [?]; digital libraries for the design and analysis of nonlinear systems [?], and scalable digital libraries containing very large amounts of multi-media data and supporting specialized algorithms for retrieving data by associations [?]. The design of these libraries is similar and illustrated in Figure 1. The design is modular and layered: individual digital libraries incorporate desired lightweight digital library services as required.

Our primary goal in the work described here is to provide a simple, lightweight video service for multi-media digital libraries built upon an ATM network infrastructure for use by the prototypes mentioned above. Our secondary goals for the video service are to provide real time video to the desktop, frame based indexing, non-linear editing, simple sound integration, and attribute based retrieval mechanisms.

By focusing on video service for work group applications, such as scientific data analysis applications, engineering design and analysis applications, computer-aided classroom instruction, and related applications, we felt we could keep the design of the video service layer simple and its implementation easy. By using a streamlined interface and supporting only minimal functionality, we felt that video could be incorporated into the multi-media digital library in a simple fashion. We want to emphasize again that it was not our intention to design or develop a prototype of a robust system which could provide reliable video to thousands of users, with guaranteed delivery and performance, but rather to understand issues related to providing a video "light" service to current and planned multi-media digital libraries.

Specifically, we imposed the following requirements:

> *Large video stores.* Many file systems today require that files be less than 2 Gbytes in size, which is unacceptable for video data. We imposed the requirement that our videostore handle data stores at least 1 Tbyte in size. This allows the videostore to provide access to workgroups of thousands of video clips.
>
> *Attribute based retrieval.* A lightweight video service should provide attribute based retrieval of data. Using this service provides a firm foundation for non-linear editing, frame based indexing, and association queries.
>
> *Manipulation of video data.* Stores of video data must support not only attribute based retrieval, but video specific functionality such as forward and reverse browsing and pausing.

3 Related Work

- For an overview on design and delivery of video service systems for multimedia applications using high bandwidth technologies, see [**?**], [**?**], and [**?**].
- For related work in video serving using relational database technology, see [**?**].
- For related work in video serving using proprietary architectures, see [**?**] and [**?**].
- For further reading in attributed based indexing and retrieval, see [**?**].

4 Design

Our approach was to allow lower level digital library services to handle basic data management and data delivery issues so that the video service per se could have a simple design. In particular, a lightweight data management layer using persistent object stores can easily handle the creation, access and updating of video data; and ATM networking technology can resolve delivery issues such as workload, congestion, and timing requirements, by exploiting built-in services for bandwidth allocation (Quality of Service QoS) and flow control.

Furthermore, because of the high-bandwidth potential of ATM, work group applications using a lightweight video service do not require many of the optimizations which are important in video delivery over lower bandwidth legacy technologies. To summarize, our approach was to view a lightweight video service as a simple application of lightweight data management and ATM networking technology.

Our lightweight data management layer is middleware that sits on top of the file system to provide the ability to access large amounts of distributed video data and retrieve it by attribute [**?**]. The use of such a middleware not only simplifies data management and distribution but also provides independence and transparency from the file system. This allows the video service to be highly portable while *increasing* simplicity.

In general one expects a trade-off between functionality and overhead: the more functionality provided the higher the overhead. Our approach is to provide lightweight protocols supporting minimal functionality and provide modular interfaces to components supporting greater functionality and requiring greater overhead.

As an example, relational and object-oriented databases also transparently manage storage resources and provide attribute based access to data. On the other hand, they also support a great deal of additional functionality, making it harder to scale them to support access to very large amounts of data, especially for continuous media applications. Lightweight object managers, however, are well suited to historical (non-changing) data, including multi-media and experimental scientific data that is contained in the digital libraries discussed above. Because of their low overhead, lightweight object managers can scale to terabyte size data sets without compromising I/O and CPU performance [**?**].

Because the lightweight data management layer uses persistent object technology, the video service can make use of object associations to combine video, audio and text. Since the data is in collections of objects, rather than streams, the video and voice can be easily time stamped, indexed and synchronized to support integration.

Finally, we decided to use the World Wide Web (WWW) as our front end, due to the wide availability of WWW browsers.

5 Implementation

Our implementation of a lightweight video service for multi-media digital libraries utilizes a persistent object data manager called PTool [?] and [?] which is compliant with a proposed standard for lightweight object management [?]. The Mpeg_play developed at the University of California, Berkeley [?], was used to initially decode legacy mpeg formatted video clips into collections of viewable frame objects that were then stored by PTool.

A World-Wide-Web based front end was developed for the video server.

In our video service implementation, PTool, the lightweight object management middleware, is used to easily manage distributed collections of viewable video frame objects. They are *viewable* in that, as objects, they have all the attributes necessary and a display method to view the video frame on an X11 served display.

The video service models a video clip as consisting of a Header object which contains the number of frames and dithering information and sequence of Frame objects. Each Frame consists of a payload of data ready for XWindows viewing. Each object Store of the video service contains a number of clips.

The video service has two simple phases. The first phase takes raw mpeg, converts it into the object model stated above, and then stripes the objects (down to the Frame level) across n nodes using PTool. In our case we striped across no more than four nodes.

The second phase simply takes the name of a clip and the nodes the movie is striped across as parameters. The viewer then requests the clip from the PTool servers running on those nodes. Once the viewer has received a frame, it calls a frame method function to view the Frame object.

ATM technology was chosen because of its very high bandwidth, the scalability of switched technologies, the ability to demand a certain Quality of Service (QoS), and its design for transportation of heterogeneous traffic (i.e. video, voice and data).

6 Experimental Results

Our experimental results were run on a four-node cluster of IBM RS/6000s connected with both ATM and 10 Mbps shared ethernet. The ATM technology used was a 2.6 Gbps FORE ASX-200 ATM switch and FORE MCA-200 Network Interface Cards for the workstations. Each switch to node connection was at OC-3 speed (155 Mbps).

We performed two separate kinds of timings. First, we timed the speed of video delivery to a single client *before* viewing. In other words, we timed the speed of video delivery independent of the machine dependent overhead of the XWindow based video display. Second, we timed the speed of video clip display on a single RS/6000 client being served by four nodes over ATM and ethernet.

PTool Version 2.1 was used for the the lightweight object management layer. ATM connections were managed by SPANS, FORE System's Switched Virtual

Circuit (SVC) ATM signalling standard.

6.1 Video Delivery Timings

When testing the *before* display timing of video delivery, we varied the number of serving nodes from one to four and varied the transmission medium between ATM and ethernet. In particular, we striped 0.74 Gbytes of video data first across one node; we then measured the access time for one client requesting the data over ATM and ethernet. Next we striped the same data across two nodes and measured the client access time, and so on. We ran the above test over both ATM and ethernet. The results are presented in Tables 1 and 2.

We discovered a linear speed-up in access times for both ATM and ethernet when increasing the number of serving nodes from one to four. Access speeds for ATM varied from two times faster than ethernet for one serving node to four times faster than ethernet for four serving nodes. The best performance measurements were obtained with four serving nodes over ATM which delivered 0.74 Gbytes of video objects in 2 minutes and 29 seconds at a speed of 39 Mbps.

6.2 Video Viewing Times

To test the video display performance of our lightweight video service, we used PTool to stripe five clips of video, at the frame level, across four video serving nodes managed by PTool. We then used the PTool/ATM based video service to request the clips and display them in an XWindow on an IBM RS/6000 with 32 Mbytes of RAM, standard 8-bit displays w/standard graphics adaptors. We compared this access and display with a PTool/ethernet version of the video service. The results are summarized in Tables 3 and 4.

It was found that the access of each clip over the ATM network was three to six times faster than that of ethernet. In all cases the access and display speeds via ATM delivered frames were close to or *faster* than real time, and ethernet access speeds were always slower than real time.

As presented in Table 4, we found that very small video clips (less than 4 Mbytes) had slower access speeds in megabits per second than larger video clips over ATM. This is attributed to a larger fraction of the overall access time being consumed by the ATM SVC connection setup time for very small video clips. This hypothesis is supported by the lack of such a discrepancy in the ethernet access speeds (see Table 4).

6.3 Pre-display vs. With-display Video Access

By comparing Tables 2 and 4, we note that with-display video access from four serving nodes is generally 2.5 times slower than pre-display access from four serving nodes for both ATM and ethernet using standard XWindows graphics. This suggests that our design of a lightweight video service is performance bound by the type of graphical display mechanism on the client host and not the delivery performance of the lightweight object management layer of the video service. Therefore, the current lightweight object management layer (PTool) allows us to improve the performance of our lightweight video service by more than a factor of two just by using a higher performance graphics client node.

7 Conclusion

There is an emerging trend to combine lightweight service layers when developing highly integrated, interactive, multi-media applications. Our lightweight video service is designed to be integrated into multi-media digital libraries. The video service itself is built upon a lightweight object management layer and a straight forward ATM networking infrastructure. Its simplicity in design and depth in functionality are good examples of the power in integrating lightweight, high performance, components for application development. In particular, we have taken a simple, yet powerful, data management paradigm (PTool) and a very flexible networking architecture (ATM) to develop a video service that is easily interoperable with a standard, platform independent front end (WWW) in order to support the production of multi-media digital libraries.

About the Authors

Shitij Mutreja is a graduate student in the Department of Electrical Engineering and Computer Science, University of Illinois at Chicago and is currently a research assistant at the Laboratory for Advanced Computing. His Internet address is mutreja@lac.eecs.uic.edu.

Stuart Bailey is a network engineer for the Laboratory for Advanced Computing; Department of Mathematics, Statistics and Computer Science; University of Illinois at Chicago. His Internet address is sbailey@lac.eecs.uic.edu.

Prof. Robert Grossman, Department of Mathematics, Statistics and Computer Science, University of Illinois at Chicago, is the director of the Laboratory for Advanced Computing. Is Internet address is grossman@uic.edu.

Dave Hanley is a graduate student in the Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago and is currently a research assistant at the Laboratory for Advanced Computing. His Internet address is dhanley@lac.eecs.uic.edu.

All authors can be reached at:

Laboratory for Advanced Computing
Department of Mathematics, Statstics, and Computer Science
University of Illinois at Chicago
851 South Morgan Street
Chicago, IL 60626
Phone: (312) 413-2176
Fax: (312) 996-1491

References

[1] S. Bailey, R. Grossman, and D. Hanley, "A Standard API For Light Weight Object Management: Strawman Version," *Laboratory for Advanced Computing Technical Report Number 95-R16,* University of Illinois at Chicago, 1995.

[2] Bucci, G., et al., "Sharing Multimedia Data Over a Client-Server Network," *IEEE Multimedia,* Fall 1994, pp 44-55.

[3] Cherkassky, V., Rooholamini, R., "ATM Based Multi-media Servers," *IEEE Multimedia,* Spring 1995, pp 39-52.

[4] Doganata, Y. N., Tantawi, A. N., "Making a Cost - Effective Video Server," *IEEE Multimedia,* Winter 1994, pp 22-30.

[5] Glauert, T., Wray, S., "Networked Multimedia: The Medusa Environment," *IEEE Multimedia,* Winter 1994, pp 54-63.

[6] R. L. Grossman, X. Qin, W. Xu, and N. Araujo, "Managing Physical Folios of Objects Between Nodes in Clusters," *Proceedings of the Sixth International Workshop on Persistent Object Systems,* Springer-Verlag, Berlin, 1994.

[7] R. L. Grossman, X. Qin, D. Valsamis, W. Xu, C. T. Day, S. Loken, J. F. MacFarlane, D. Quarrie, E. May, D. Lifka, D. Malon, L. Price, "Analyzing High Energy Physics Data Using Databases: A Case Study," *Proceedings of the Seventh International Working Conference on Scientific and Statistical Database Management,* IEEE Press, 1994, In press.

[8] R. L. Grossman, D. Hanley, and X. Qin, "PTool: A Light Weight Persistent Object Manager," *Proceedings of SIGMOD 95,* ACM, 1995, In press.

[9] R. Grossman, X. Qin, and W. Xu, and H. Hulen, and T. Tyler, "An Architecture for Scalable Digital Libraries," *Proceedings of the 14th IEEE Computer Society Mass Storage Systems Symposium,* S. Coleman, editor, IEEE, 1995, In press.

[10] R. L. Grossman, A. Nerode, and W. Kohn, "Nonlinear Systems, Automata, and Agents: Managing their Symbolic Data Using Light Weight Persistent Object Managers," *Proceedings of FGCS'94 Workshop on Heterogeneous Cooperative Knowledge-Bases,* Kazumasa Yokota, editor, Springer-Verlag, In press.

[11] Iwata, A., Mori N., et al., "ATM Connections and Traffic Management Schemes for Multimedia Internetworking," *Communications of the ACM,* February 1995 V38 N2, pp 73-88.

[12] Little, T. D. C., Venkatesh, D., "Prospects for Interactive Video-on-Demand," *IEEE Multimedia,* Fall 1994, pp 14-24, pp 73-89.

[13] Smith, B. C., Rowe, L. A., "A Continuous Media Player," *Proceedings of the 3rd International Workshop on Network and OS Support for Digital Audio and Video,* San Diego, CA, November 1992.

[14] Smoliar, S. W., Zhang, H., "Content-Based Video Indexing and Retrieval," *IEEE Multimedia,* Summer 1994, pp 62-72.