Yima: Real-Time Multimedia Storage and Retrieval

Kambiz Ghahremani Cyrus Shahabi Shu-Yuen Didi Yao Roger Zimmermann
University of Southern California
Los Angeles, CA 90089
(213) 740-7654
{ghahrema, shahabi, didiyao, rzimmerm}@usc.edu

ABSTRACT
Yima is a scalable, real-time streaming architecture that enables applications such as video-on-demand and distance learning on a large scale. While Yima incorporates lessons learned from first generation research prototypes, it also complies with industry standards in content format (MPEG-4) and communication protocols (RTP/RTSP). Yima improves upon both research and commercial approaches by using a *bipartite* design and alternative approaches to handling variable-bit-rate (VBR) video. We also integrated a selective retransmission protocol into Yima's RTP server to recover from packet loss. Lastly, we tweaked available hardware and software to achieve certain objectives (e.g., playback of HDTV streams on an HDTV monitor). Yima is operational and supports a variety of display bandwidths.

Keywords
Continuous media, video server, scalable, MPEG-4, and multimedia storage.

1. INTRODUCTION
A growing number of applications such as news-on-demand, distance learning, e-commerce, corporate training, and scientific visualization store, maintain, and retrieve large volumes of real-time data, where the data are required to be available online. We denote these data types collectively as *continuous media*, or CM for short. Continuous media is distinguished from traditional textual and record-based media in two ways. First, the retrieval and display of continuous media are subject to real-time constraints. Second, continuous media objects are large in size. Popular examples of CM are video and audio objects, while less familiar examples are haptic, avatar and application coordination data [1].

Several commercial implementations of continuous media servers are now available in the marketplace. We group them into two broad categories: 1) single-node, consumer oriented systems (e.g., low-cost systems serving a limited number of users) and 2) multi-node, carrier class systems (e.g., high-end broadcasting and dedicated video-on-demand systems).

Commercial systems often use proprietary technology and algorithms, therefore, their design choices and development details are not publicly available and objective comparisons are difficult to achieve. We designed and developed a second generation continuous media server, called *Yima*. Yima incorporates and combines useful lessons learned from first generation research prototypes in terms of random or constrained data placement, non-deterministic or restricted time-period scheduling, and variable or constant bit-rate media. The server is designed as a completely distributed system which supports scalability, fault- tolerance, and heterogeneous disk subsystems. Yima is compatible with industry standards including MPEG-4 and the RTSP/RTP communications protocols.

Several interesting innovations of Yima are worth noting: 1) complete distribution where all nodes run identical software and where single points of failure are eliminated, 2) efficient, online scalability of disks where disks can be added or removed without stream interruption, 3) synchronization of several independent streams of audio and/or video within one frame (one-thirtieth of a second), and 4) smooth, client-dictated transmission rate control.

2. SERVER DESIGN
An important component of delivering isochronous multimedia over IP networks to end users and applications is the careful design of a multimedia storage server. The task of such a server is twofold: (1) it needs to efficiently store the data and (2) it must schedule the retrieval and delivery of the data precisely before it is transmitted over the network. We incorporate a *bipartite* design for our Yima server which offers multiple connections between the server nodes and the client.

There are two basic techniques to assign the data blocks to the magnetic disk drives that form the storage system: in a *round-robin* sequence [2], or in a *random* manner [3]. Traditionally, round-robin placement utilizes a cycle-based approach for the scheduling of resources to guarantee a continuous display, while the random placement utilizes a deadline-driven approach. Random placement has several advantages such as 1) support for multiple delivery rates with a single server block size, 2) simpler design of the scheduler, 3) support for interactive applications, and 4) automatic achievement of the average transfer rate with disks that are multi-zoned. Finally, random placement results in more efficient data reorganization when the system scales up (or down) through a technique called SCADDAR [4].

Scalability is addressed with several techniques for CM servers. Instead of a single disk, multi-disk arrays are employed.

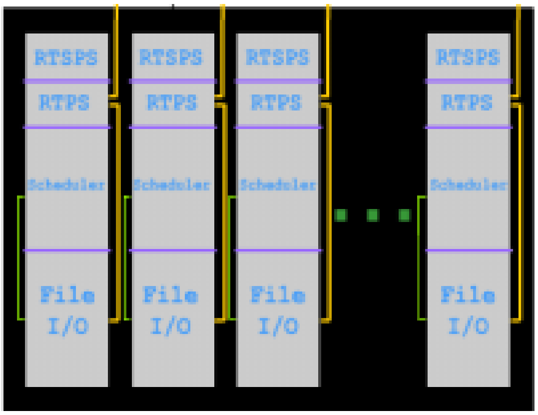

Figure 1: Components of a 4-node Yima server.

However, if all the disks are connected to a single large computer then the I/O bandwidth constraints of this computer will limit the overall achievable throughput, hence multiple computers, or multi-nodes, are used. The storage nodes are interconnected via a high-speed network fabric that can be expanded as demand increases. This modular architecture provides a cost-effective and scalable solution in which older PCs can be easily replaced by newer, more advanced PCs. More nodes can be added to support more simultaneous video requests. Figure 1 shows the components of a 4-node server. Applications, such as video-on-demand, that rely on large-scale CM servers, require continuous operation around the clock. To achieve high reliability and availability for all the CM data that are stored in the server, Yima implements a parity-based data redundancy scheme that, in addition to providing fault-tolerance, can also take advantage of a heterogeneous storage subsystem through a technique called *Disk Merging* [5].

3. CLIENTS

We have built a client application called the *Yima Presentation Player* to demonstrate and experiment with our Yima server. The Yima player can display a variety of variable bit-rate media such as MPEG-2, MPEG-4, and HD content on both Linux and Windows platforms. We have even designed a client that supports the synchronized playback of 5 channels of panoramic video along with 10.2 channels of surround sound audio. The client constantly monitors the level of data in its buffer to determine whether to issue a speed-up or slow-down command to the server. Using this method, the client is able to tune the amount of data in its buffer.

4. RTP AND PACKET RETRANSMISSION

Yima supports the industry standard real-time protocol (RTP) for the delivery of time-sensitive data. To reduce the number of lost RTP data packets we implemented a selective retransmission protocol [6]. When multiple servers deliver packets which are part of a single stream the following question arises: If a data packet did not arrive, how does the client know which server node attempted to send it? There are two solutions to this problem: 1) the client computes which server node to issue the retransmission request to, or 2) the client broadcasts the retransmission request to all the server nodes. Yima incorporates the unicast model where retransmissions are sent to the correct server nodes.

5. CONCLUSION

Yima is a second generation scalable real-time streaming architecture, which incorporates results from first generation research prototypes, and is consistent with industry standards. To eliminate bottlenecks, we use a bipartite design which also results in close to linear scale-up. Yima is a working prototype with high-end clients, an efficient communication protocol with retransmissions, and support of variable bit-rate video.

We conducted several extensive sets of experiments. We sent MPEG-4 data from the Yima servers in our lab to the public Internet via the USC campus network. The client was set up in a residential apartment and linked to the Internet via an ADSL connection. We also performed experiments in high-speed WAN environments. We conducted recent tests via a trans-continental SUPERNET link from the Information Sciences Institute (ISI East) in Arlington, VA, to USC in Los Angeles, CA. We successfully demonstrated the synchronized transmission and playback of High Definition MPEG-2 video and 10.2 channels of uncompressed, immersive audio at a total data rate of 55 Mbps. Information about these experiments, which we term Remote Media Immersion (RMI), is available on the USC Integrated Media Systems web site (imsc.usc.edu/rmi) and the SUPERNET Next Generation Internet (NGI) web site (www.ngi-supernet.org).

We are exploring a more distributed architecture of Yima which would allow a wider range of serviceable clients. We will also be extending the support of other data types such as haptic and avatar data.

6. REFERENCES

[1] C. Shahabi, G. Barish, B. Ellenberger, N. Jiang, M. Kolahdouzan, A. Nam, and R. Zimmermann. Immersidata Management: Challenges in Management of Data Generated within an Immersive Environment. In *Proceedings of the International Workshop on Multimedia Information Systems*, October 1999.

[2] S. Berson, S. Ghandeharizadeh, R. Muntz, and X. Ju. Staggered Striping in Multimedia Information Systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1994.

[3] J. R. Santos and R. R. Muntz. Performance Analysis of the RIO Multimedia Storage System with Heterogeneous Disk Configurations. In *ACM Multimedia Conference*, Bristol, UK, 1998.

[4] A. Goel, C. Shahabi, S.Y. D. Yao, and R. Zimmermann. SCADDAR: An Efficient Randomized Technique to Reorganize Continuous Media Blocks. In *Proceedings of the International Conference of Data Engineering*, 2002.

[5] R. Zimmermann and S. Ghandeharizadeh. Continuous Display Using Heterogeneous Disk-Subsystems. In *Proceedings of the Fifth ACM Multimedia Conference*, pages 227–236, Seattle, Washington, November 9-13, 1997.

[6] C. Papadopoulos and G. M. Parulkar. Retransmission-Based Error Control for Continuous Media Applications. In *Network and Operating Systems Support for Digital Audio and Video*, Zushi, Japan, April 23-26 1996.