The Case For An Integrated Multimedia File System

Pawan Goyal, Sriram S. Rao, Prashant J. Shenoy, and **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: {pawang,sriram,shenoy,vin}@cs.utexas.edu

Recent advances in computing and communication technologies have led to the development of applications that involve storing, accessing, and processing multiple forms of information (such as audio, video, images, text, etc., collectively referred to as multimedia data). Traditional file systems have been optimized for the storage and access of textual data, and hence, are inefficient at managing audio and video data. On the other hand, specialized file servers developed for continuous media data (referred to as video-on-demand servers) do not support the storage and retrieval of textual/numeric data. Since it is cumbersome to support multimedia applications in environments consisting of multiple file servers - each maintaining a different data type, it is essential to design an integrated multimedia file system that can efficiently manage multiple data types.

Due to the fundamental differences in characteristics between conventional and continuous media data, no single policy is adequate for meeting the requirements of all data types. For instance:

- Whereas a small block size provides adequate performance for textual data, recent research efforts have shown that a large block size is required to optimize performance of the server for audio and video. Consequently, the storage space manager must be able manage multiple block sizes within a single file system.
- To facilitate editing operations (such as inserting a frame into a video file), the meta data structure must provide mechanisms for indexing media units (such as video frames or audio samples) in addition to the traditional byte stream interface.
- Multimedia requests need real-time bounds on the response times, while good response times without any absolute guarantees suffice for textual data.
- Due to the large storage space requirement of multimedia data, most multimedia file systems are founded on disk arrays. Whereas the inherent redundancy in video streams can be exploited to approximately reconstruct lost image data due to a disk failure (and thereby impose smaller overhead on the disk array during failure recovery), perfect recovery is necessary for textual data.

Hence, an integrated multimedia file system must support multiple policies for efficient management of multimedia data as well as provide a framework for their coexistence. This requires significant changes to the architecture of a file system. A file system that meets these requirements is being built at the Distributed Multimedia Computing Laboratory, UT-Austin.