Multimedia file serving with the OS/390 LAN Server

by M. G. Kienzle
R. R. Berbec
G. P. Bozman
C. K. Eilert
M. Eshel
R. Mansell

The rapidly increasing storage and transmission capacities of computers and the progress in compression algorithms make it possible to build multimedia applications that include audio and video. Such applications range from educational and training videos, delivered to desktops in schools and enterprises, to entertainment services at home. Applications developed for stand-alone personal computers can be deployed in distributed systems without change by using the client/server model and file servers that allow the sharing of applications among many users. The OS/390™ LAN Server has been enhanced to support multimedia data delivery. Resource management and admission control, wide disk striping to provide high data bandwidths, and multimedia-specific performance enhancements have been added. The resulting server benefits from the robustness, scalability, and flexibility of the S/390® system environment, which allows it to move into new multimedia applications. Multimedia support on a robust, widely installed platform with little or no additional hardware requirements gives customers the opportunity to enhance their existing applications with multimedia features and then expand their capacity as the demands of the applications increase. This multimedia server platform is in use with several interesting applications.

The rapid increase in processing, storage, and network transmission capacities of computers, combined with ever lower cost, has made possible the use of multimedia data such as video and audio in computer applications. In addition to augmenting existing applications, many new types of applications are becoming possible. Examples of such applications are interactive multimedia education, just-in-time training, multimedia-based information services, and video-on-demand services to homes. In some instances, such as with services to the home, a new transmission infrastructure is needed. In other cases, existing infrastructure such as file servers, computer networks, and client computers are in place and can be augmented gradually as the applications' capacity demands increase.

Many multimedia applications have been developed for stand-alone use on personal computers (PCs) or UNIX** workstations, which we will call client computers. They store the multimedia data on CD-ROM or on local disks. These applications control all real-time aspects directly through the application code, or through a special middleware layer. They obtain the data through the file system from the disks. The part of the application program that moves the data in real time is often called the *player*.

The easiest way to port applications from a stand-alone execution environment into a client/server

0018-8670/97/$5.00 © 1997 IBM

environment is through file server technology. Instead of using a local file system, file operations are redirected to a remote file server. All permanent data of an application, the executable code as well as the multimedia files, can reside on the file server. The execution of the application, including the decompression of compressed video files, occurs on the client computer. The redirection of file operations is performed by code in the client computer, often called a *redirector*, and involves no change in the application code.

File server technology has been in place for some time. The most widely used file server protocols are: the SMB (server message block) protocol[1] used by IBM's Operating System/2* (OS/2*) LAN Server[2] and by Microsoft's LAN Manager**,[3] the Novell NetWare***[4] NCP** (NetWare Core Protocol),[5] and the NFS** (Network File System**) protocol[6] that is most popular in UNIX environments but has also found use in other workstation systems. File servers allow sharing of files by many users, eliminating the need for private copies of files. This is particularly important for video files that are very large even when compressed.

Video applications and their contents are usually expensive to create. This makes it desirable to share them among as many users as possible. Their size also makes them expensive to store, so a single copy of the material is advantageous. A single copy of the data also makes management tasks such as updates, backups, and access control easier. Clearly, the client/server paradigm makes sense for multimedia files.

Mainframes have long been known as robust platforms for I/O-intensive applications requiring industrial-strength data management. They easily scale up to meet large capacity requirements in both storage and throughput. With the recent introduction of CMOS (complementary metal-oxide semiconductor) hardware, mainframe cost, size, and power requirements are comparable to the alternatives.

The OS/390* LAN Server[7] is part of the OS/390 system. When a multimedia server for MVS (Multiple Virtual Storage) was needed, the OS/390 LAN Server, a very high-performance file server for workstation clients, was enhanced to support multimedia serving. This meant extending the OS/390 LAN Server implementation to improve throughput for multimedia read operations, to add bandwidth management and admission control, and to increase the maximum file size. The data blocks are distributed across multiple disks on a block basis, with a technique called *disk striping*, to allow a large number of video streams to be created from a single copy of the data. The OS/390 LAN Server is now in use with a diverse set of multimedia applications.

This paper describes the design of the OS/390 LAN Server's multimedia features. First, we describe the basic OS/390 LAN Server, designed to be used as file server for a wide range of workstation clients. Next, we examine the environment in which multimedia file systems operate, as well as their requirements. Then we describe the architecture and design of the multimedia extensions. After presenting some server configurations and performance data, we give examples of how customers use the OS/390 LAN Server to serve multimedia files in a client/server environment.

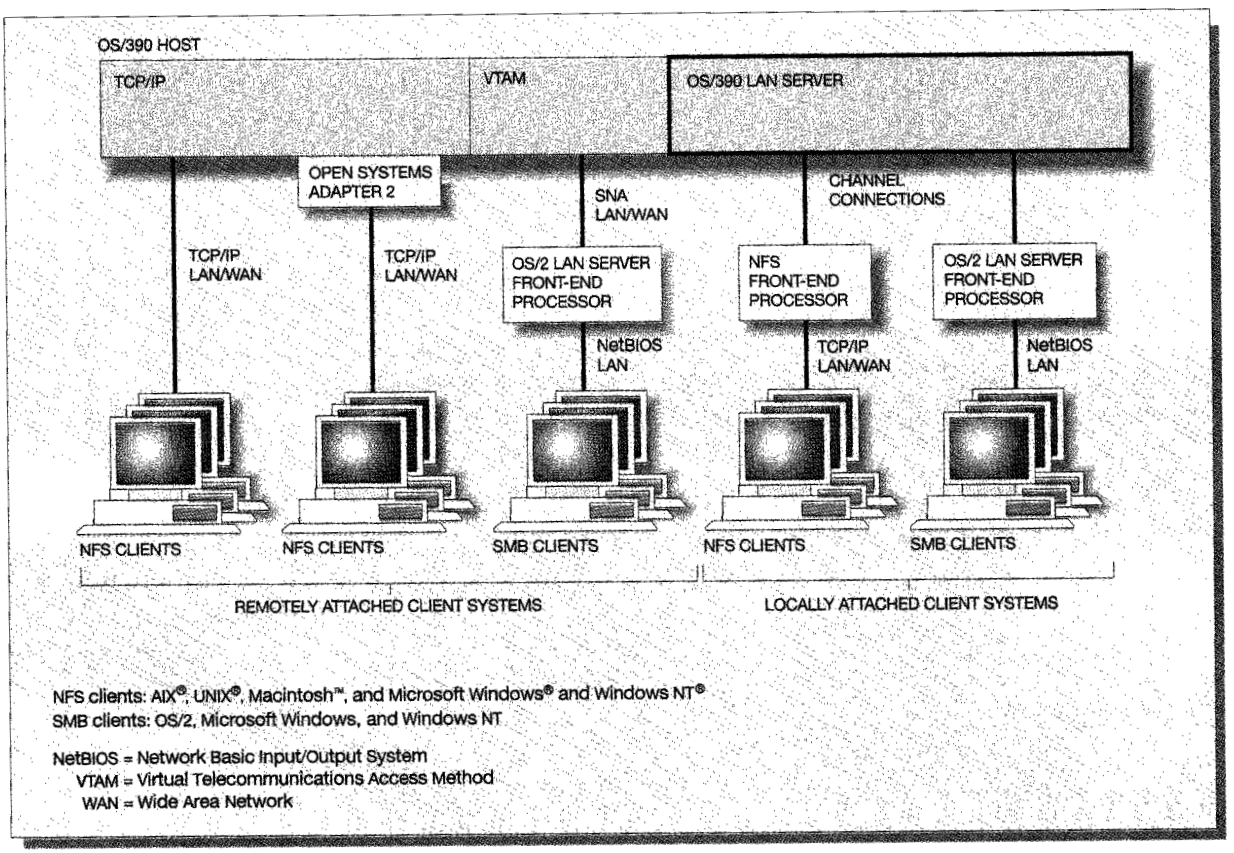
Basic OS/390 LAN Server architecture

The OS/390 LAN Server is a high-performance file server running on MVS/ESA* (Enterprise Systems Architecture), serving data to personal computers and UNIX workstations. It supports very large numbers of clients[8] simultaneously, integrating islands of smaller PC-based file servers into one large file system image. A single file system image for a large number of clients provides substantial advantages in storage cost, access management, and system management. The OS/390 LAN Server supports simultaneous shared access to files on the server through any of the major client/server file access protocols: SMB, NFS, and, through the IBM Research prototype described in this paper, Novell NCP.

Figure 1 shows the end-to-end software configuration of an OS/390 LAN Server domain. SMB and NFS clients access the server through different paths. All clients are standard clients and require no change to access the OS/390 LAN Server.

The OS/390 LAN Server architecture supports intermediate servers, which are also called *front-end processors*. The OS/390 LAN Server, the front-end processors, and the clients form a three-tier client/server infrastructure. The objective of using front-end processors is to off-load function that does not have to be performed in the OS/390 LAN Server, and to avoid duplicate implementation of function already implemented on the front-end processors. At the same time, the front-end processors improve the OS/390 LAN Server's performance by caching data blocks in RAM (random access memory) and by man-

Figure 1 OS/390 LAN Server end-to-end configuration



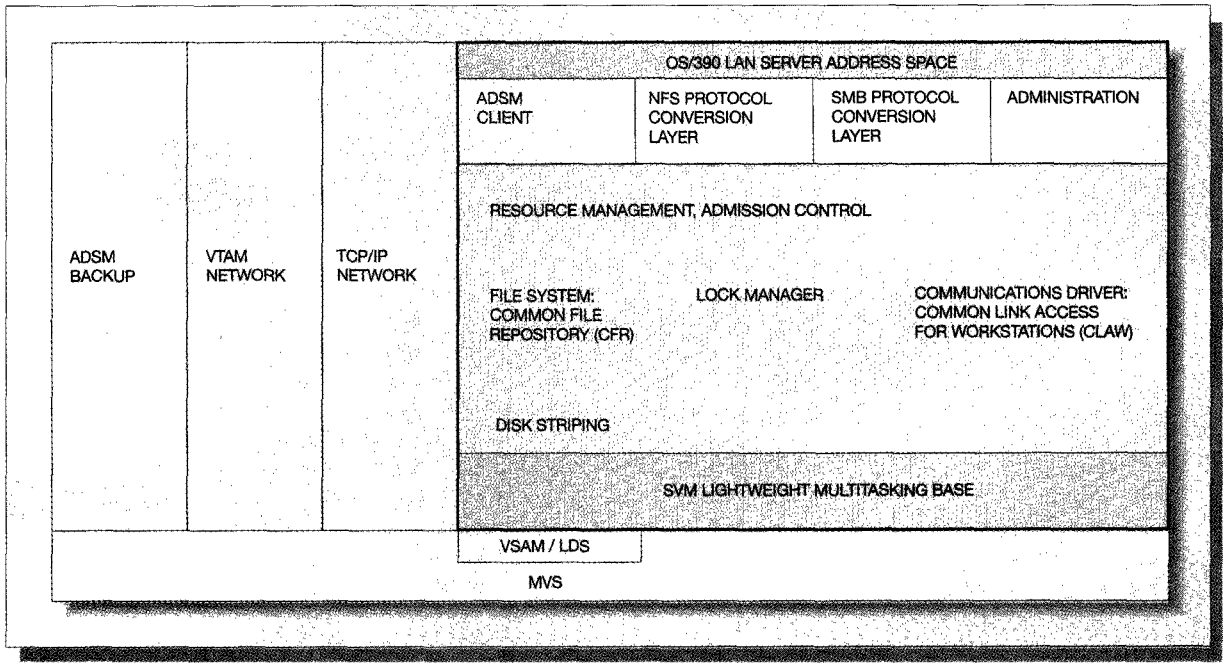
aging the LAN communication with the clients. Finally, they allow users to store on the front-end processors files that need not be shared across the entire OS/390 LAN Server domain. This three-tier infrastructure gives the users flexibility to design the most efficient client/server architecture for their requirements. There are two major ways of attaching the client systems: *local connections* from the OS/390 LAN Server over S/390* channels attached directly to the front-end processors to provide high-performance access; *remote connections* using general network protocols such as TCP/IP (optionally through an Open System Adapter) or SNA to reach clients over a wide area network.

OS/390 LAN Server software structure. The OS/390 LAN Server software structure is highly modular, supporting a variety of file server protocols and communications protocols. Figure 2 shows the internal structure of the OS/390 LAN Server.

The OS/390 LAN Server uses the *Common File Repository* (CFR) as its file system. CFR supports the combination of file system functions required by SMB and NFS. It is a modern high-performance file system implementing features such as hierarchical directories, atomic metadata updates to eliminate file system consistency checks, byte-range locking, hard and soft links,[9] extended attributes, and arbitrarily long file names using a full character set. CFR takes advantage of the efficient implementation of *linear data sets* (LDSs) under VSAM (Virtual Storage Access Method) by mapping its logical disks onto VSAM linear data sets.

A set of common services provides the infrastructure on which the *protocol conversion layers* (PCLs) are implemented. The SMB PCL implements the SMB file server protocol, the NFS PCL implements the NFS version 2 file server protocol, and the administra-

Figure 2 Internal software structure of the OS/390 LAN Server



tion PCL performs the administrative tasks of setting up and operating the server.

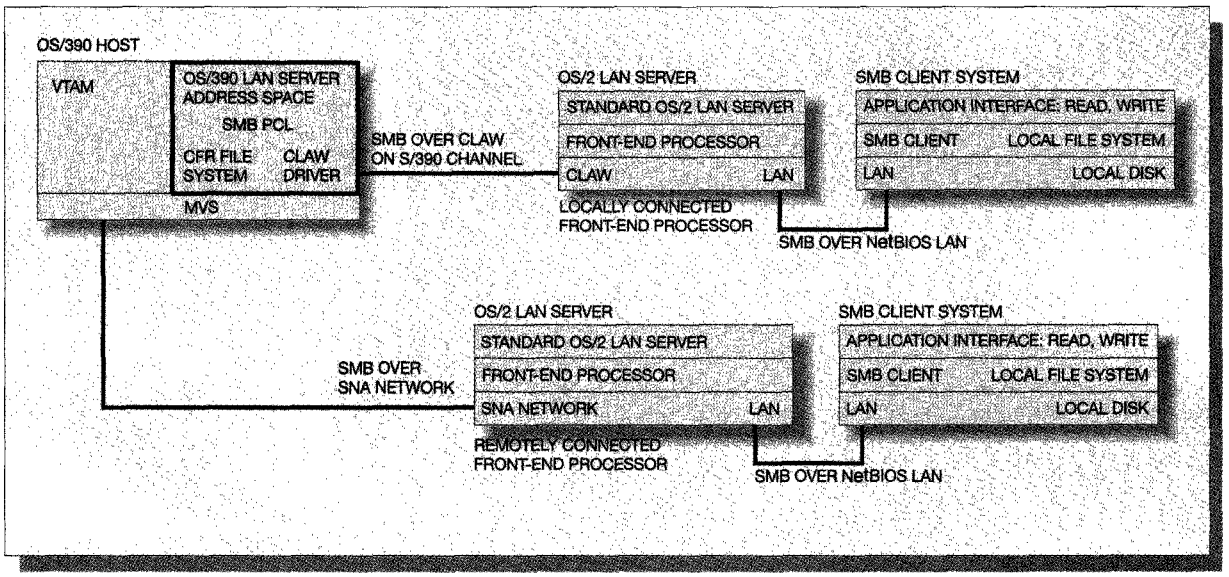
The *common link access to workstations* (CLAW) function supports locally attached front-end processors. CLAW establishes connections to the front-end processors, either over ESCON* (Enterprise Systems Connection) channels or over S/390 parallel channels, and uses an optimized "lightweight" protocol for communication. This high-throughput, low-response-time connection contributes to the excellent performance characteristics of the OS/390 LAN Server in local environments.

The *lock manager* supports a general file-locking protocol, as well as data integrity across multiple block caches in the front-end processors. For file locking, the lock manager implements a superset of the lock protocols associated with the file server protocols supported by the OS/390 LAN Server. For instance, the SMB protocol requires byte-range locking, whereas the NFS protocol utilizes only advisory locking. The lock manager also keeps read locks on the cache contents of the front-end processors. When a client modifies a file block that is resident in one or several front-end processor caches, the lock manager issues "cache invalidate" signals to these front-end processors, assuring data integrity. This approach avoids having to broadcast the signals to all front-end processors, which would cause overhead for the front-end processors that do not have the updated blocks in their caches.

The OS/390 LAN Server uses the ADSTAR* Distributed Storage Manager (ADSM) Server[10] to back up its files. The *ADSM client* establishes the connection between the OS/390 LAN Server and the ADSM Server. The back-up function is contained entirely on the S/390 system and occurs without any interaction with the client systems. In fact, the client systems do not even need to be connected for their files on the OS/390 LAN Server to be backed up.

A server system has to perform many tasks, such as I/O operations, concurrently. It is important that the server software be able to efficiently share state information among all its activities. The OS/390 LAN Server runs in a single MVS address space but uses multiple task control blocks (TCBs) to utilize multiple processors. Inside each TCB, it uses the shared virtual machine (SVM) lightweight multitasking base to coordinate its activities efficiently.

Figure 3 SMB client-specific infrastructure



Client-specific infrastructure. The OS/390 LAN Server supports clients using either the SMB protocol or the NFS protocol to access files on the server. It has a separate infrastructure for each of the two protocols.

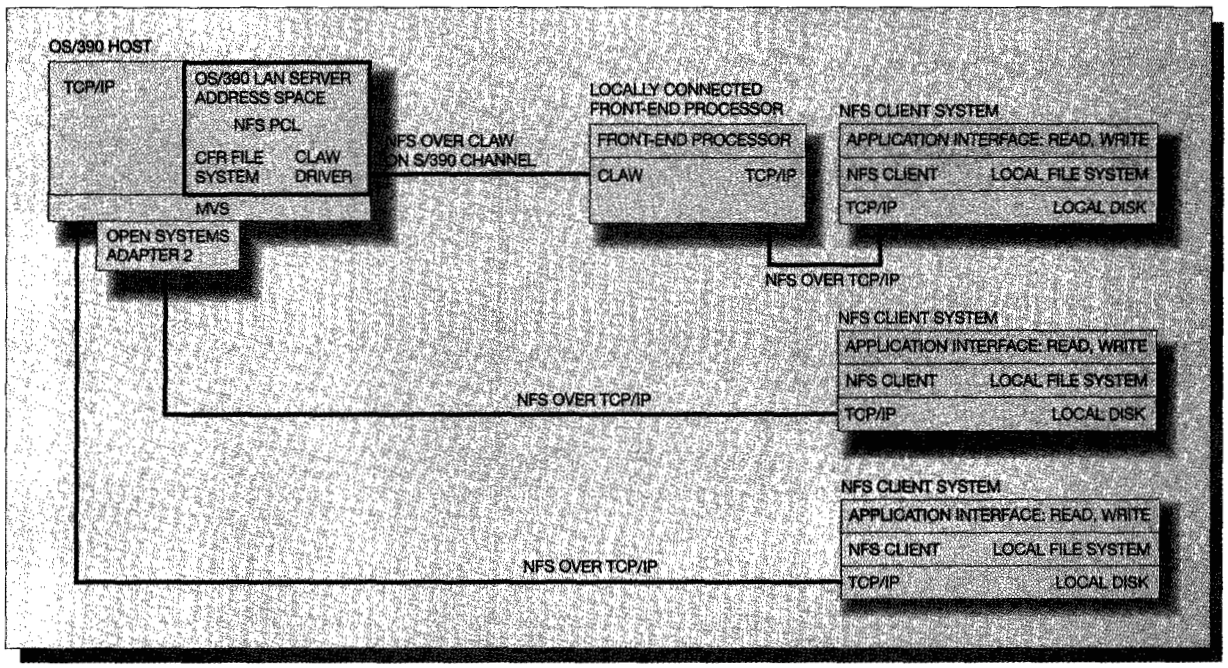
SMB clients. SMB clients are connected to an OS/2 LAN Server over a LAN using the NetBIOS (Network Basic Input/Output System) protocol. The OS/2 LAN Server is connected to the OS/390 LAN Server locally via an S/390 channel using the CLAW protocol, or remotely via an SNA connection through the VTAM* (Virtual Telecommunications Access Method) function on the OS/390 host.

Figure 3 shows the software structure in the OS/2 LAN Server when it is part of an OS/390 LAN Server system. To attach the OS/2 LAN Server to the OS/390 LAN Server, the front-end processor function must be installed in the kernel of the OS/2 LAN Server. When an SMB client wants to access files on the server, the standard OS/2 LAN Server performs client authentication and access control. The administration may place some of the files on the OS/2 LAN Server; others may reside on the OS/390 LAN Server. The client can never explicitly discern the actual location of the files.

When a client accesses a file residing on the OS/390 LAN Server, the front-end processor function in the OS/2 LAN Server intercepts the request and forwards it to the OS/390 LAN Server without further processing. This interception is performed entirely in the kernel of the OS/2 LAN Server, assuring high performance and low latency. The SMB protocol conversion layer in the OS/390 LAN Server handles the request and returns the response to the front-end processor function, which in turn sends the response to the client. The front-end processor function also contains a block cache in RAM to avoid having to pass to the host requests for recently used blocks. The lock manager in the OS/390 LAN Server is used to track the cache contents across all front-end processors to ensure file system integrity across the entire OS/390 LAN Server complex. The front-end processor always requests at least a 4 KB (kilobyte) block from the OS/390 LAN Server, which, when combined with the cache, creates "read ahead" and more efficient transfers than small sequential read operations.

NFS clients. NFS clients connect to the OS/390 LAN Server locally through an NFS front-end processor, and remotely over a wide area network using the TCP/IP protocol; the TCP/IP protocol is supported either channel-attached directly to the host, or through the S/390 *Open Systems Adapter 2* feature in the S/390 host system (see Figure 4). TCP/IP overhead on the S/390 server can be avoided by using the NFS front-

Figure 4 NFS client-specific infrastructure



end processor. This results in higher throughput and shorter response times. The NFS front-end processor is not a full-function file server, it is an intermediary processor that improves OS/390 LAN Server performance by off-loading TCP/IP protocol processing from the S/390 host and through a read-ahead cache. In particular, the NFS front-end processor can read blocks larger than the 8 KB block size that is typical for NFS. This feature reduces the number of read operations from the front-end processor to the OS/390 LAN Server. Alternatively, NFS clients connect to the OS/390 LAN Server through the TCP/IP implementation on the S/390 host. Regardless of the connection type, the OS/390 LAN Server appears to the client as any other NFS server, implementing the NFS version 2 protocol.

Multimedia file system environment and requirements

Multimedia file serving extends the traditional file server requirements in two major directions. First, for each client viewing a video, the data delivery rate, also called the *stream rate*, must be guaranteed to assure continuous, smooth delivery of video and audio data. Second, the files involved are very large, and the bandwidth required to deliver them is very high. Typical data rates are 1.5 Mbps (megabits per second) to 6 Mbps, resulting in 200 to 800 KB of data for each second of video, or about 720 MB to 2.8 GB (gigabytes) for an hour. To support many clients simultaneously viewing the same video file, the server must be able to serve the aggregate data rate of many video streams from a single file. When a single disk cannot support the aggregate data rate requirement of a particular video, additional bandwidth can be obtained by replicating the video files on more than one disk, or by striping the video file across multiple disks. Clearly, disk striping is the more economical solution. To support high video throughput the server must use all system resources involved, such as disk bandwidth, central storage bandwidth, and CPU processing power, as economically as possible.

On a personal computer, the user controls all applications and can make sure that sufficient system resources are available to move the real-time data at the appropriate rate from the local disk via the decompression hardware (or software) to the display and the speakers. Since a file server is shared by many users, it has to ensure explicitly that each video stream has sufficient resources available for smooth transmission to the client workstation. This requires new software on the server that guarantees bandwidth to a real-time data stream and rejects real-

time file requests for which the delivery rate cannot be guaranteed.

High-quality video applications are expensive to produce and can be justified only when they are being used by large audiences. When users have their own data copy, either the storage cost is very large, as with conventional hard disks, or the storage cannot be updated easily, as with CD ROMs. A single server supporting a large number of viewers from a single copy of the data on hard disks can readily address these problems of storage cost and data management.

Existing computer networking software generally focuses on reliably transferring files but does not provide performance guarantees. Therefore, for multimedia files, network resource management and reservation software must be added.

Many applications do not explicitly know the bandwidth requirements of the material they are playing. No standard software protocol exists to request a particular bandwidth guarantee. In a multimedia client/server application, the server has to know or assume bandwidth requirements and perform admission control and resource allocation accordingly.

Multimedia files are generally very large compared to the amount of RAM storage available for file-block caching on server systems. Therefore, traditional file-block caching frequently results in a poor cache "hit" ratio for multimedia files. Furthermore, putting a file in a cache usually requires an additional data copy at a time when it is not known whether that copy will be used. Thus, traditional file-block caching is not effective in multimedia file servers.

Design of the OS/390 LAN Server's multimedia features

The efficient support of multimedia data required modifications in many areas of the OS/390 LAN Server. These modifications consist of performance optimizations, resource management functions, and new infrastructure components, such as the metafiles that contain control information needed by the resource management.

High-performance features of the Common File Repository. At the outset of the multimedia server project, a number of file systems were considered, including a custom real-time file system to be developed "from scratch." Analysis showed that CFR was an excellent starting point and would meet the functional and performance goals with some extensions that were consistent with its basic design. This section outlines some of CFR's features that proved important in the performance analysis.

CFR is designed to run on lightweight threads that handle each incoming request from beginning to end. Using its multiprocessing-safe lock manager, these threads execute in multiple processes that can be dispatched to run on multiple processors, achieving a high degree of parallel operation. To reduce the number of I/O operations and interrupts, CFR will always try to chain blocks onto existing CLAW I/O operations.

CFR depends minimally on operating system services. In particular, CFR itself provides frequently used services that it can implement at lower cost than the operating system. For instance, the CFR has its own free-storage manager, using the "binary-buddy" algorithm[11] with double-word increments, for storage requests of up to 4 KB. Larger amounts of storage are obtained from the MVS free-storage manager.

CFR is optimized for data chunks of 4 KB increments. This is the size of the disk blocks normally cached in virtual storage by the data cache manager. Although caching is not efficient for multimedia files, it is still used for small files, metafiles, and for file system metadata like the pointer blocks used in the process of sequentially reading a large file. The cache directory uses a two-level discontiguous coalesced hash table. The first level of the directory has a pointer to each discontiguous second-level page. The cache is dynamic and therefore the directory hash table can grow and shrink. New second-level pages are obtained from dynamic storage whenever a page grows beyond its current "high watermark." Storage is returned whenever the directory is reduced by 50 percent.

CFR keeps metadata such as directories and attributes in a cache that is independent of the file block cache. A monitor collects statistics and balances the available free storage with the cache size. The monitor swaps out metablocks only when the data cache has shrunk to zero and when free storage is at its "low watermark."

The disk block allocation table is split between the metablock section and the data block section so that metablocks, except for pointer blocks, are not interleaved with data blocks. A moving cursor is used to

make sure that data blocks and pointer blocks are allocated sequentially.

Large data block transfer. The high data rates of multimedia streams make low overhead and high throughput one of the most important goals for a multimedia server. Reading large blocks from disk amortizes the fixed cost of a disk operation, the seek and latency times, over a large amount of data. Since a large portion of the I/O overhead is independent of the size, large data transfers save processing time for both the disk I/O and the network I/O operations. For all these reasons, the OS/390 LAN Server transfers large data blocks from disk to central storage, and from central storage to the CLAW driver.

Large data transfers save processing time for both the disk I/O and the network I/O operations.

However, at the start of a read sequence the OS/390 LAN Server reads shorter blocks as requested by the client. This is done, first, to determine whether the client will read the file sequentially when read-ahead is economical, and, second, to avoid a startup delay. In an ongoing read sequence, delay is hidden by the double-buffer scheme; this permits large blocks to be read with low overhead. Longer blocks can lead to longer queuing times and consume more buffer space. The efficiency of long blocks must be balanced with the responsiveness of shorter blocks. Read block sizes of 540 KB for disk transfers and 60 KB for CLAW transfers have proven to be good compromises in multimedia applications.

Shared buffer pool, caching, and fast buffers. Copying data as they move through the system can often be a large source of processing overhead in data transfer applications such as file serving. Typically, data are read from disk into buffers owned by the operating system and are then copied to the application's address space. Likewise, on the path into the network the data are usually copied from application space into system buffers, and from there to the I/O channel. Data caching may result in an additional copy.

The reasons for these copy operations are manifold. First, it is more efficient to manage system buffers without having to consider when the application is finished with the data. Second, giving application programs access to system buffers makes it more difficult to achieve system security. Finally, segmentation and the addition of headers, and reassembly and removing of headers, is easier when the data are copied.

In file servers, there is no reason to copy the data into the application's address space, since the file server does not process the data. Furthermore, the file server code is trusted and is inaccessible to the users, so the security exposure does not exist. Therefore, using a shared buffer pool does not violate security and it is vital to minimize the processing overhead and the associated storage bus usage in server applications.

The multimedia support in the OS/390 LAN Server eliminates all data copies in the host on data paths that use channel-attached front-end processors by having the CFR file system and the CLAW communications driver share a buffer pool. At the startup of the OS/390 LAN Server, the buffer pool is allocated and fixed in central storage, as is required for I/O buffers. The SMB PCL moves the data from CFR to CLAW by handing off pointers. When CLAW is finished with the buffers, it marks them. CFR inspects the marks when it needs new buffers, so no explicit signaling between threads is necessary.

When clients are connected through wide area networks implemented on the host, this simple buffer-sharing scheme does not work. Instead buffer sharing between the OS/390 LAN Server and the VTAM or TCP/IP address spaces is needed. As this has not been implemented yet, multimedia serving using these protocols is not as efficient as using locally attached front-end processors.

Multimedia files are very large compared to file block caches. Unless one viewer is reading a video file just after another, it is not likely that a block cache will experience many cache hits. Moreover, the implementation of the block cache implies a data copy. Since multimedia files are read sequentially most of the time, prefetching the file blocks sequentially from disk using double or triple buffers is more effective than caching. Therefore, the OS/390 LAN Server of-

fers such a buffering option, which it calls *fast buffers*. In the fast-buffer mode the file system allocates a set of buffers to an open file, and as soon as a buffer is empty the file system issues a read-ahead operation to fill that buffer again. That is, the server tries to keep the buffers full at all times. This mode of buffer use is called *greedy prefetch*.[12] The rate at which buffers are read from disk is determined by the rate at which blocks are requested by the client systems. The server operates in *pull mode*; that is, the application in the client system controls the rate and timing of the data transfer. Aggressive prefetching means that whenever a new block read request arrives at the server, the data are in the server's buffer. Effectively, the read request is synchronous. This assures low latency for sequential read requests; it also smoothes out variations in the data rate. Greedy prefetch at the client system can also cover up variations in the network delay.

Disk striping. Multimedia files require stream rates between 200 and 800 KB/second. Current disks can sustain aggregate data rates of up to 10 Mbps. Therefore a disk can support only a fairly small number of simultaneous real-time data streams. To support additional viewers, another copy of the material can be placed on a different disk. This, however, will result in higher storage costs. A better solution is to stripe the video files across many disks. Here, files are allocated such that sets of data blocks are distributed in a round-robin fashion to a set of disks known as a *striping group*. The *striping granularity*, that is, the amount of sequential data on one disk, should be the same as the read block size. This way, each read operation will result in one I/O operation to one disk. Reading a file sequentially results in interleaving accesses across all disks of the striping group. This layout of data on disk is sometimes called *block interleaving*. Reading multiple files simultaneously results in statistical multiplexing of disk accesses across the striping group. With striping, each disk has to support only a fraction of the bandwidth for a particular stream, and therefore can support a much larger number of streams. On average, all disks in the striping group experience the same bandwidth demand.

The OS/390 LAN Server implements *wide* disk striping entirely in software. It uses the linear data sets, each of which is mapped to a physical disk, as disk surrogates, and defines striping groups across multiple LDSs. An individual file can be as large as 4 GB. The maximum number of LDSs in a striping group is 256, providing a very large pool of disk bandwidth to support many video streams. Each member LDS of a striping group can be up to 4 GB. The CFR file system views each of these striping groups as a single logical disk for purposes of block allocation and bandwidth management. Special recovery operations that are used in true RAID[13] (redundant array of inexpensive disks) systems are not supported for the striping groups.

Striping files across disks is commercially available, mainly through RAID disk controllers. However, RAID controllers are only a partial solution for the bandwidth problem with multimedia files. Commercially available RAID controllers usually support "4 + 1" or "8 + 1" disk configurations, resulting in striping group sizes of 4 or 8. In many cases, this is not wide enough for multimedia serving.

Resource management and admission control. For smooth flow of the video data through the server, the bandwidth of the server's resources must be managed explicitly to avoid overload and interruption of video and audio streams to the client systems. *Resource management* determines the maximum bandwidth of the server resources, allocates them to video streams, and rejects new streams if insufficient bandwidth is available for them.

Resource management in the OS/390 LAN Server manages the bandwidth of logical disks (single disks or striping groups) and of channel bandwidth to the front-end processors. For each resource, the configuration file specifies a maximum bandwidth. For logical disks, a *calibration* process can determine this bandwidth dynamically.

Whenever a new multimedia file is opened for real-time viewing, resource management checks to determine whether there is sufficient bandwidth left on the logical disk where the file resides and on the channel through which the front-end processor is attached. If there is more than one replica of the file, resource management selects the most suitable replica for opening. If there is sufficient bandwidth, resource management reserves the bandwidth and allows the file to be opened. If there is insufficient bandwidth, the client is told that the file cannot be served now.

Non-real-time access to the data is supported through background I/O operations. The OS/390 LAN Server does not begin a background I/O operation unless there is unused disk bandwidth. This restriction, along with the administrator limit on the num-

ber of background tasks, will ensure that non-real-time data streams do not compromise real-time data streams.

Resource management assumes that it has exclusive use of the resources it manages: the disks, disk controllers and channels, and the CLAW channels. If this is not true, resource management may not be effective. The CPU resources are allocated by the MVS dispatcher. The OS/390 LAN Server must be given high enough dispatch priority to assure availability of CPU processing power in the presence of other applications.

Calibration. The calibrator determines the bandwidth that a particular disk or striping group of disks is capable of supporting. It provides the maximum bandwidth value that resource management uses when allocating bandwidth and when placing video files onto disk. Every time the disk configuration changes, the calibrator should be invoked. It starts a number of streams, typically six to twelve per disk. Each stream reads blocks as quickly as it can. The calibrator determines the maximum bandwidth capacity by dividing the total number of bytes read by the duration of the calibration process. The result of the calibration process may be optimistic. The calibration addresses only the bottleneck from the disk subsystem into the server; it does not include the data movement from the server to the front-end processors. The resources used for the front-end operation are more predictable, and so, calibration is generally not necessary.

Data placement. Placing video files on logical disks is a complex problem. For traditional data only the storage capacity must be considered. For multimedia data, the aggregate bandwidth that a video file is expected to support must be considered as well. Resource management in the OS/390 LAN Server places video assets onto the disks or striping groups based on the expected popularity of the video. The administrator who adds a video file to the system is asked to supply the *expected viewers* (EV), the number of simultaneous viewers of this video during the heaviest viewing period. The EV value multiplied by the viewing data rate yields the peak bandwidth for that video. The sum of the peak bandwidths of all videos on a disk is the peak bandwidth of the disk.

Resource management uses the bandwidth-to-space ratio (BSR) algorithm[14] to select a disk for the new video. It bases the selection on the size and the peak bandwidth of the video to be loaded, and the current peak bandwidths and space utilization of the eligible disks. The expected peak bandwidth of all videos on a disk or a striping group may exceed the bandwidth capacity of the disk. If the peak bandwidth of a single video exceeds the bandwidth capacity of a disk, resource management will create more than one replica. Because the peak bandwidth of video disks is used for data placement, it is important to keep the EV values of the videos current.

Resource management places video assets on logical disks based on the expected popularity of the video.

A further problem of data placement occurs at a lower level, where the CFR file system decides which physical blocks on the disk are to be used to hold a file. The block allocation algorithm in CFR attempts to keep all of a file's data blocks together and separate from those of other files. When a file is created, the space it requires is, in essence, set aside on the disk. If another file is created by some other task a very short time later, its space will be assigned to follow that set aside for the first file.

Metafiles. To support data management of the multimedia files and resource management, the file server has to store new kinds of data about a file, such as the bandwidth at which the file has to be played, and the number and location of the replicas of the video file. This information is stored in the *metafiles*. Video files are not directly visible to clients; instead, the client sees the metafile for each video file. When a client attempts to open a metafile, the file server selects a replica, opens the actual data file, and gives the "handle" for the actual file back to the client. From then on, all read operations use the file handle of the actual file.

To make sure that metafile operations and general file requests do not interfere with the real-time playing of video files, the OS/390 LAN Server manages three types of disks: PWS (programmable workstation) disks, metadisks, and video disks. PWS disks hold client data without real-time requirements. The metadisk, of which there can be only one, holds the

metafiles. The video disks hold the actual video files. Access to the video disks is controlled by resource management to avoid overload.

To enter a file onto the metadisks and video disks, it must first be written to a PWS disk. A special command transmits the real-time attributes of the file, another command invokes the data placement algorithm, creates the metafile, and moves the file onto the video disk. If specified by the data placement algorithm, it also creates multiple copies to satisfy the peak bandwidth requirement.

Extended attributes. The way in which CFR maintains extended attributes associated with a file is designed to benefit the performance of large files. Since many file reads occur in page multiples beginning at a page boundary, I/O operations for data on page boundaries are particularly efficient. CFR preserves the first 64 KB of a file's allocation space for extended attributes (the total length of all extended attributes cannot exceed 64 KB). The file's actual data begin after that. This causes most files to be sparse, since the physical disk space assigned to the extended attributes depends on their actual length. When the extended attributes are changed, the file's data do not have to be moved. In addition, it is guaranteed that the file's data begin on a page boundary for efficient I/O operations.

CLAW communications driver. The CLAW communications driver was enhanced in several ways to support multimedia file serving. It shares a buffer pool with the CFR file system to avoid data copies. A dual-priority algorithm for scheduling data transfers on the S/390 channels to front-end processors assures a smooth flow of real-time data in the presence of other file requests from the OS/390 LAN Server. First, CLAW gives read requests for video data priority over read requests for PWS disks. This isolates the video traffic from the impact of regular file requests. Second, CLAW gives a real-time video operation priority over a non-real-time operation. This isolates real-time requests from the impact of background I/O operations.

As a result of this strict priority implementation, non-real-time files can be starved when the entire bandwidth on a CLAW link is allocated to real-time operations. This can be avoided by reserving some bandwidth on the CLAW link for non-real-time operations. It is essential to regulate non-real-time requests because they are usually copy or move operations. These operations have no rate control and can consume the entire disk bandwidth. Note that this priority is effective only for the OS/390 LAN Server data traffic within the CLAW context.

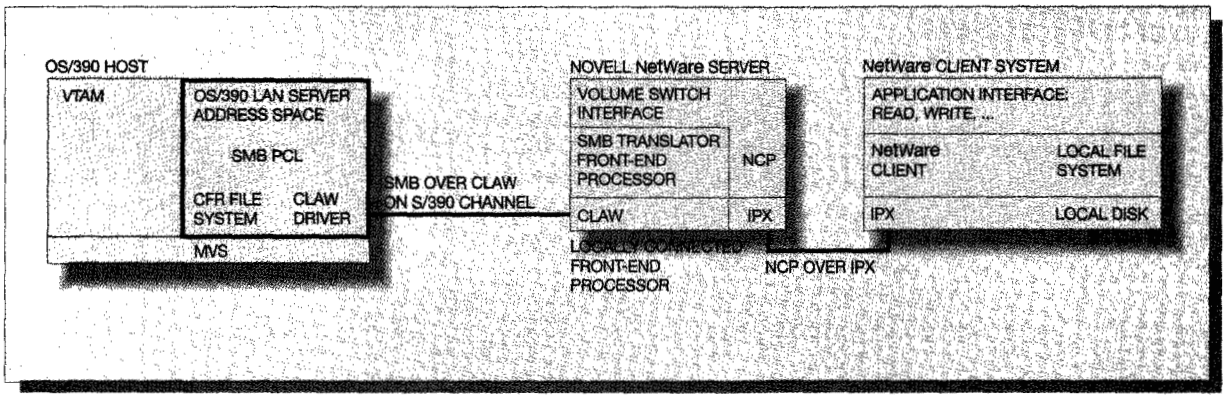
Multimedia support on the OS/2 LAN Server front-end processor. The OS/2 LAN Server Ultimedia* must be installed on the OS/2 front-end processors in order to support multimedia serving via the SMB protocol. This multimedia version of the OS/2 LAN Server incorporates a Resource Reservation System (RRS) for management of real-time resources on the front-end processor. When the OS/2 LAN Server Ultimedia is used as a stand-alone multimedia server, the RRS manages the network bandwidth to the clients and the disk bandwidth of the OS/2 LAN Server. When used in conjunction with the OS/390 LAN Server, the RRS manages only the network bandwidth to the clients. Resource management in the OS/390 LAN Server manages the remaining real-time resources.

The front-end processor code in the OS/2 LAN Server also employs the fast-buffer mode and a greedy prefetch algorithm when it transfers video files. For real-time read requests it rounds up to larger block sizes, typically 60 KB, for a request to the S/390 host. Since the clients may use any block size for read requests, rounding up the requests and prefetching data preserves the server's efficiency and assures short latencies without requiring any changes in the client software. To avoid the uncontrollable latencies that a LAN bridge would introduce, it is recommended that clients are attached through single-segment LANs to the OS/2 LAN Server. However, all bandwidth can never be effectively used; testing indicates that 5 is the maximum number of clients on a 10 Mbps Ethernet segment that can receive 1.5 Mbps streams.

Multimedia serving: Pull mode vs push mode. One of the most important features of a multimedia system is the smooth flow of the data to the player application. This flow is maintained through careful pacing of the data, based on the requirements of the application. Since clock management across several systems in a wide area network is generally difficult, the pacing is controlled by only one clock.

If the controlling clock is in the client system, it controls the data flow and pulls data across the network. To avoid "jitter" due to server and network latencies, the client employs a greedy prefetch algorithm. This mode of operation is called the *pull mode*. It requires that the pacing signals, typically block read requests, be transmitted from the client to the server

Figure 5 Infrastructure for the IBM Research prototype supporting Novell NetWare front-end processors



with little delay. Using pull mode over a standard file system interface at the client system, any application on a client can use video files on the server. Furthermore, the greedy prefetch algorithm and the pacing by the client can easily deal with variations in the data rate of some types of compressed video files, and with variations in the network latency. It also handles application-induced changes in the data rate such as pause, slow motion, or still frame. Pull mode frees the server from detailed knowledge of video formats and instantaneous data rates.

In cases where the network is not bidirectional and does not allow pacing from the client, the server's clock controls the data flow and pushes the data through the network to the client. This is called the *push mode*. Under push mode, the client transmits VCR (videocassette recorder) commands such as play and pause to the server, and the server moves the video data until it receives the next command from the client. The control commands are typically transmitted over a separate out-of-band connection. The control lag between the client and the server can result in a fairly complex command implementation. For video files with variable data rate the server must inspect the data to determine the frame boundaries, so as to keep the frame rate constant. Application-controlled changes in the data rate require special handling by the server. Latency variations in the network may still require substantial buffering in the client.

The push mode is preferred in a variety of network environments, such as when the network's ability to transmit pacing signals to the server is extremely limited, when the channel is asymmetric, as with cable television distribution systems, or when the client has little buffer space and processing capability, as with cable television set-top boxes. Streaming video on the World Wide Web is another widespread use of push mode. A comparison of the push and the pull modes has been published.[12] When the client systems have a reasonable amount of RAM for buffering, and when the network has sufficient back channel capacity, it is usually easier to implement a pull model. The pull model is also more easily adaptable to support existing stand-alone applications that run on client computers. The OS/390 LAN Server product currently supports only the pull mode. A prototype front-end processor has been developed that supports a push model for specially configured client systems.

Prototype extensions to support Novell NetWare clients

In response to customer requests, IBM Research has developed a prototype extension that allows client systems such as OS/2, Microsoft Windows** 3.1 and Windows 95**, UNIX, and Macintosh** systems to use Novell NetWare client software to share data on the OS/390 LAN Server. A major design goal of this prototype was to support the delivery of multimedia streams to all NetWare client systems. This extension required changes in the OS/390 LAN Server code and that the new front-end processor code be embedded in a Novell NetWare server as a NetWare loadable module. Figure 5 shows the Novell NetWare-specific infrastructure.

OS/390 LAN Server extensions. Support by the OS/390 LAN Server for clients connected to a Novell NetWare server is provided via extensions to the standard SMB protocol conversion layer. In addition, the prototype supports all of the file-serving functions expected of a general-purpose file server. These extensions provide multiple name spaces and Macintosh file system support that is not present in the standard SMB protocol.

Multiple name spaces. Novell NetWare provides a name-mapping function that makes it possible for users to see the names of files on a NetWare server according to the rules of their client operating system. That is, the same physical file may have several different names for different client systems. For example, DOS imposes a restriction that its file names must be in the "eight-dot-three" format, whereas the Macintosh operating system permits up to 31 characters. Also, the character sets permitted in a file name vary between client operating systems.

When a file is created on a NetWare server, NetWare assigns names to it that obey the naming conventions of other client operating systems. A similar function has been added to the OS/390 LAN Server for users connected via a Novell NetWare server. DOS and Macintosh users are assigned their own name spaces; all other users share the *default* name space.

Macintosh file system support. The Macintosh file system has certain unique characteristics. The most important of these is the concept of file "forks." These may be thought of as distinct streams of data associated with a given file, each of which can be accessed and manipulated independently of the others.

The Macintosh file system implements two such streams called the "data fork" and the "resource fork." The data fork contains the file's actual information, whereas the resource fork contains information that may describe an application's menus, dialog boxes, icons, and many other things, perhaps even including the executable code of an application itself. In addition to the file forks, the Macintosh file system maintains a unique set of metadata.

The extensions to the OS/390 LAN Server support five streams of information and extended metadata for Macintosh clients connected via Novell NetWare. The first data stream, called the primary data stream, contains the file's data; the second stream is used for the Macintosh resource fork; a third will be used to contain access rights information, leaving the remaining two for possible future expansion.

NetWare front-end processor. To complement the OS/390 LAN Server extensions, IBM Research also developed an extension to the NetWare operating system, a NetWare loadable module (NLM), which connects a NetWare front-end processor to an OS/390 LAN Server. OS/390 LAN Server directory trees are manifested to NetWare clients in the same manner as NetWare-managed logical volumes, and, in terms of function and performance, NetWare clients cannot distinguish between *local* and *remote* volumes.

A NetWare 4.1 interface called the *volume switch* was employed to redirect file system requests to a channel-connected OS/390 LAN Server via a CLAW communications driver. This interface is used by the NetWare 4.1 NFS Gateway. The volume switch is close to the top of the LAN protocol stacks, and the parameters it uses are closely related to the NetWare Core Protocol (NCP) file system requests that flow over the LAN in IPX (Internet Packet Exchange) packets. Based on the volume specified in the incoming request, the front-end processor calls either the internal NetWare service routine (for local volumes) or a service routine registered by the NLM (for remote volumes).

Unlike OS/2 LAN Server front-end processors, the NCP-like format of the volume-switch interface requires translating its protocol to the Server Message Block (SMB) protocol before requests can be forwarded to the OS/390 LAN Server, and transforming its responses to NCP before control is returned to NetWare. While in the simplest cases this translation is straightforward, in many instances it is complex (e.g., one to many or many to one). This complexity requires the IBM Research NLM to maintain state in order to convert NetWare objects to OS/390 LAN Server objects and vice versa.

The IBM Research NLM employs aggressive read-ahead and lazy-writing strategies to improve the performance of larger files. Directory information that has been recently referenced is cached in RAM, and a synergistic interface to the NetWare block cache is being developed to improve the performance of frequently referenced "small-" to "medium-" sized files.

The IBM Research NLM is designed to be easily extended to connect to multiple, possibly heterogeneous, hosts that support the SMB protocol. The components

that support protocol translation, communication, caching, etc., are sufficiently independent that the NLM might be extended to allow bridging to a host that supports a protocol other than SMB.

Multimedia support for the Novell NetWare Server. Client systems connected via Novell NetWare are able to benefit from the multimedia extensions added to the OS/390 LAN Server. Specifically, files may be read using the fast-buffer facility. The resulting ability to rapidly and smoothly feed large multimedia files to NetWare client systems is particularly important in educational environments where Macintosh processors are often used.

As more and larger interactive multimedia applications are being developed, it is desirable that they be stored at a single location that has the ability to deliver them efficiently to many end users. The OS/390 LAN Server, with its Macintosh file system support and its high-performance interface to Novell NetWare servers, is an ideal repository for such applications.

System configuration and performance

The installation of an OS/390 LAN Server to support a multimedia workload requires special care to ensure smooth operation. In particular, the system must be configured to meet the capacity requirements. Also, the performance management parameters must be set to support the resource management function.

System requirements and server capacity. The streaming capacity of the OS/390 LAN Server can be up to a thousand streams, depending on the stream rate and the speed and the number of processors of the S/390 system deployed as servers. The storage capacity of the OS/390 LAN Server is limited only by the number of disks that can be installed and dedicated to multimedia file serving.

The processing cost for the OS/390 LAN Server to deliver data from disk to the channel is approximately one instruction per byte. This relationship is essentially independent of the data rate of the individual streams. For example, one stream at 800 KB/second will require approximately the same processing power as four concurrent streams at 200 KB/second. Essentially, the aggregate data rate capacity increases linearly with the processing power. The OS/390 LAN Server accomplishes this by spreading work across multiple MVS task control blocks (TCBs) and, consequently, across multiple processors in an S/390 Central Electronic Complex. The OS/390 LAN Server creates one set of TCBs to move data from disk to central storage and a second set of TCBs to move the data into the network. The maximum effective parallelism would be achieved by configuring one disk access TCB per S/390 processor and one network-serving TCB per front-end processor.

As multimedia serving is a novel application, only limited performance data are available. In one case, an IBM S/390 dual-processor model 9121-480 with operating system MVS 5.1 running an OS/390 LAN Server delivered 80 video streams, each at 400 KB/second, at 80 percent CPU utilization. The 80 streams of video were delivered from a single striped data set that spanned 18 disk volumes. The 18 disk volumes were spread over three 9345 direct access storage subsystems with six volumes each. Four ESCON channels connected the storage subsystems to the S/390 processor. The system had six 3172 front-end processors with 32 MB of memory, running OS/2.

The OS/390 LAN Server uses fixed central storage buffers to eliminate data movement within central storage. The aggregate fixed central storage requirement is linear with respect to the number of streams. The fixed buffer size required depends on the disk configuration and data striping. Jitter-free video delivery has been achieved using various buffer sizes. With a pair of buffers assigned to each stream, and buffers up to 540 KB, the fixed central storage requirement can be up to 1080 KB per stream.

Performance management. Resource management controls the I/O bandwidth from the disks and to the front-end processors. To achieve jitter-free video delivery, the OS/390 LAN Server address space must have sufficient and timely access to the processor. If video serving is the only workload on the system there is no competition for the processor. However, it is not necessary to dedicate a system to video serving when using the OS/390 LAN Server. The MVS operating system is designed to run mixed workloads with different responsiveness requirements. Using dispatching priorities, the MVS system can allocate to the OS/390 LAN Server processor resources sufficient for jitter-free video delivery while also running other workloads, such as transaction processing or batch workloads. MVS sets resource access controls, such as dispatching priorities and central storage allocations, based on customer input. In workload manager goal mode, the workload manager service policy provides this input.[15] The OS/390 LAN Server

address space should receive a high velocity goal and a high importance. Then, the workload manager will allocate the S/390 server resources to meet the OS/390 LAN Server's goals.

The MVS workload manager provides an additional feature that is useful when serving video while running mixed workloads. The workload manager "resource group maximum" feature can be used when video serving adversely impacts the other workloads on the system. The resource group maximum specifies the maximum rate at which a group of address spaces can consume processor service. The workload manager limits the service to the maximum amount specified. It does not affect address space groups that are consuming less than the maximum service specified. Consequently the high velocity, high importance goal for the OS/390 LAN Server can be used in combination with the resource group maximum to guarantee smooth, uninterrupted video and to protect other workloads.

Prototype deployments

The OS/390 LAN Server and the IBM Research NLM prototype have been deployed in some early customer studies to gain experience with video serving in a realistic application environment. This section will describe two types of environments in which the OS/390 LAN Server is being used.

Campus network: University of Nebraska. The University of Nebraska at Lincoln is using the OS/390 LAN Server and the IBM Research prototype NLM to support educational multimedia experiments at the university and at local high schools. The EduPort distance learning project uses an OS/390 LAN Server and a high-speed network to provide real-time on-demand access to digital libraries and museums. Documents, images, and full-motion videos are digitized and stored at the University of Nebraska and then sent over fiber optics facilities to Lincoln High School where teachers and students can use the information when it is most useful, on demand (see Figure 6).

For the EduPort demonstration a "super server" located at the University of Nebraska supporting a LAN server and token-ring network of clients at Lincoln High School delivers multimedia on demand over a high-speed fiber optics cable. The super server is an S/390 model 9121-621, which runs MVS Release 4.3 and has 15 GB of disk space to store the content. A channel-attached PS/2* Model 95 is used as the front-end processor. The workstation runs OS/2 LAN Server and is capable of supporting 48 concurrent video streams through four token-ring connections. The Lincoln Telephone and Telegraph Company provides the wide area network (WAN) connectivity to Lincoln High School through their LAN Emulation Services.

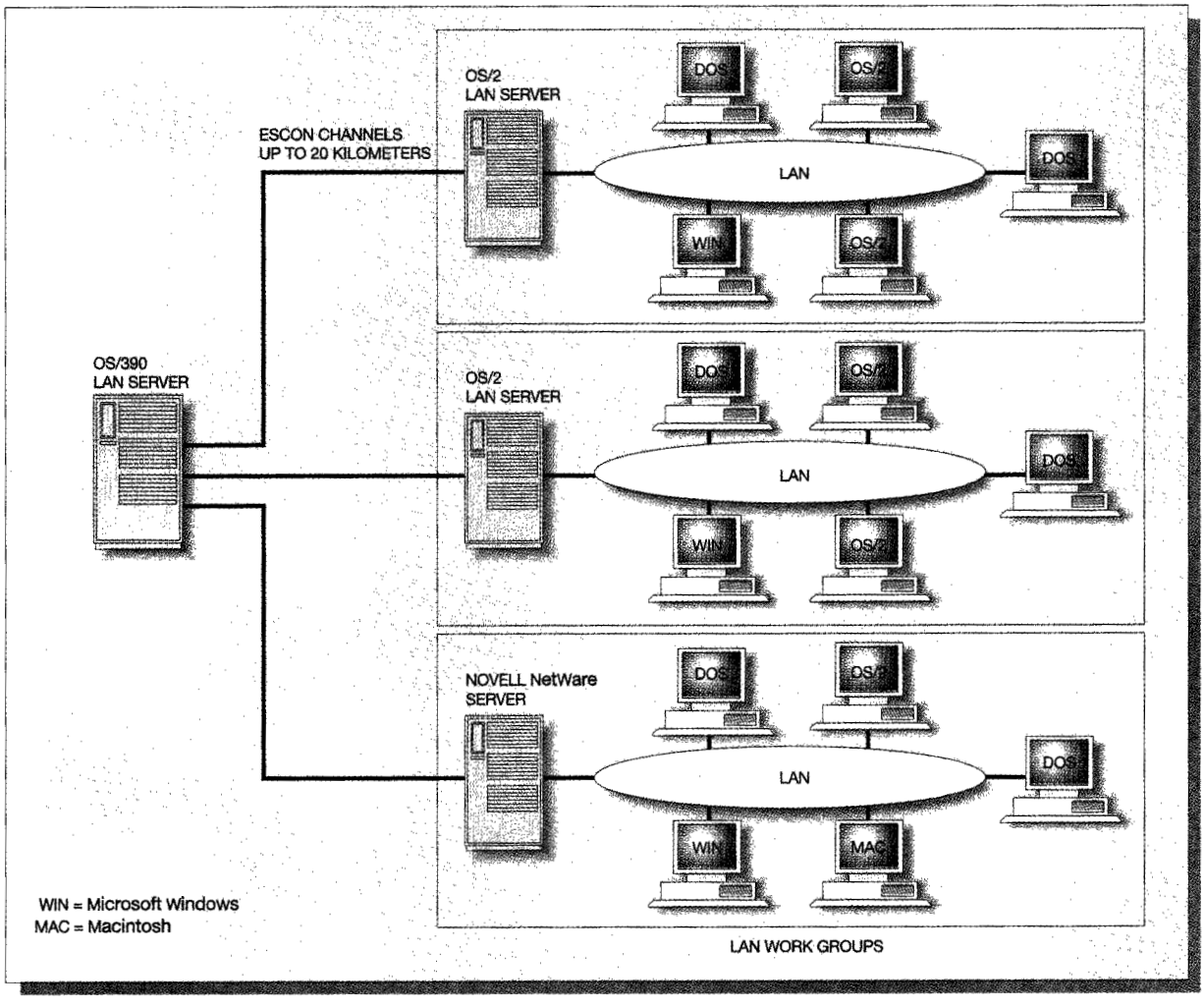
On the university campus, the NetWare front-end processor for OS/390 LAN Server is especially popular because of the extensive use of Macintosh processors and NetWare. Although definitive measurements were not available when this paper was submitted, preliminary performance results are encouraging.

Digital library projects. The OS/390 LAN Server is also being used in production digital library projects at Virginia Commonwealth University (VCU) in Richmond, Virginia, and Marist College in Poughkeepsie, New York. At each of these schools the server is an S/390 Enterprise Server Model 9672-R42 with MVS Release 5.2.2 or OS/390 Release 1. The channel-attached front-end processor is a 3172 Interconnect Controller or a PC 500* running OS/2 Warp Server. At Marist College the campus network connection is currently token ring but is being upgraded to ATM (asynchronous transfer mode). VCU's video delivery system uses a 155-Mbps connection to an ATM backbone, with 25-Mbps ATM Forum-compliant[16] LAN emulation ATM connection to workstations in selected classrooms.

At Marist College the OS/390 LAN Server is integrated with an IBM Digital Library solution as a multimedia server and LAN file server. Initial use of the video serving capabilities of OS/390 LAN Server at VCU involves an innovative case study system in the School of Pharmacy. In this case both the case study application files and MPEG-1 (Moving Pictures Experts Group standard 1) video content are served over the ATM network using the high-speed serving capacity of the OS/390 LAN Server.

Metropolitan area network. The OS/390 LAN Server supports a project that provides a "video-on-demand" system to 30 elementary and junior high schools over a metropolitan area network (MAN) in Okazaki City, Japan. The Okazaki City public school system has a long history of creating audiovisual contents. In addition to the video server, IBM provides the imaging equipment and studio support required to create the educational video material. Advanced development tools include the IBM POWER Visual-

Figure 6 Multimedia file serving on university campuses

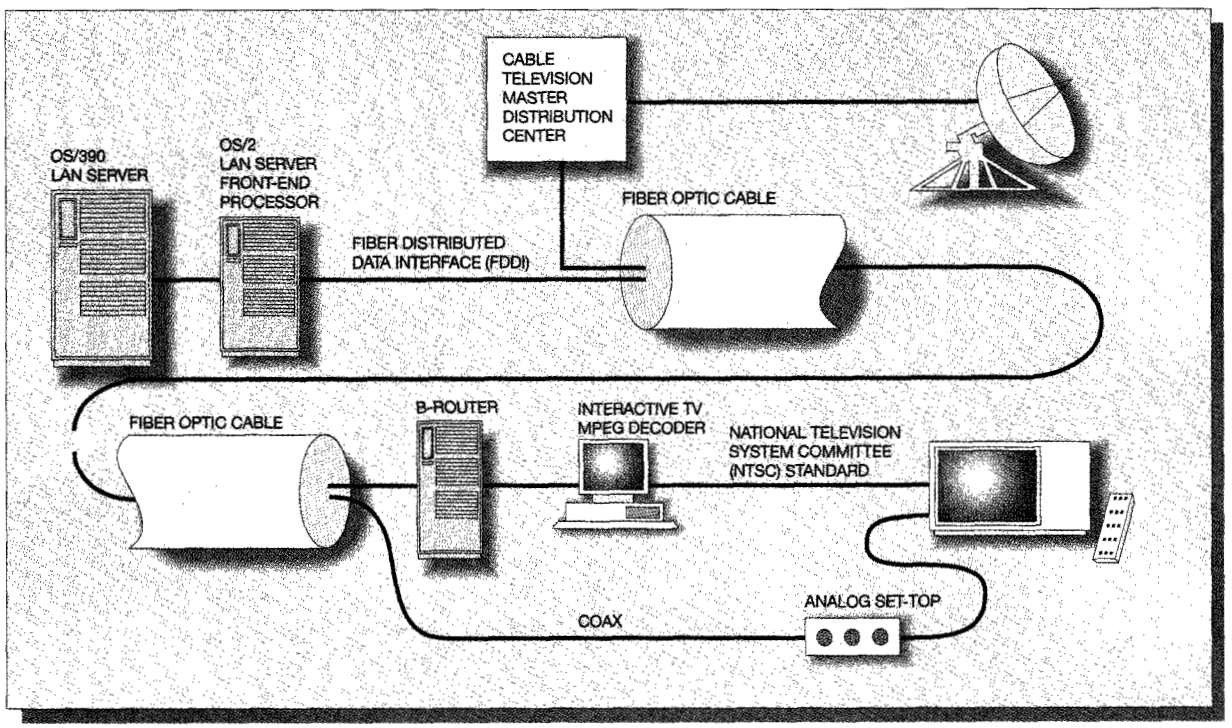


ization System* that has already been used by the Hollywood film industry to create stunning high-quality graphical material.

Digital video data are compressed and stored on the 100 GB of 9345 disk storage attached to the IBM 9672 Parallel Transaction Server. This multimedia server runs the MVS (Version 5.1) operating system, and the video files are stored and delivered by the OS/390 LAN Server (see Figure 7). The system supports 80 simultaneous video streams at 3 Mbps. IBM worked with partners to install the high-speed optical-fiber backbone network covering the metropolitan area in which the 30 schools are located. The longest distance between a school and the video server is 30 kilometers. Each school has two IBM client multimedia PCs for classroom use by teachers and children. These PCs run OS/2 Warp, and are equipped with IBM MPEG-1 adapters that are capable of decompressing video material compressed at 3 Mbps. This data rate, twice the rate usually associated with MPEG-1, was necessary to provide sufficiently high image quality.

The benefits are substantial. The teachers and children have access to a wealth of educational resources. All of this material can be simultaneously accessed by every school. Classes can be built around mul-

Figure 7 Multimedia file serving to schools in a cable MAN environment



timedia content to communicate and educate more clearly. Central management of the video library makes it easier to update all course materials.

Summary

The basic OS/390 LAN Server has been enhanced to support multimedia file serving to network-attached PCs and UNIX workstations. Many optimizations and special functions supporting multimedia files assure smooth and cost-effective operation. An IBM Research prototype using a Novell front-end processor extends these functions to all clients of Novell NetWare servers including, in particular, Macintosh clients. The OS/390 LAN Server has been successfully deployed in a great range of applications and network environments. It has proven to be an effective tool for supporting large-scale networked multimedia applications, particularly for education.

Acknowledgments

The initial prototype of the LAN server was developed by the Operating System Structures group at Research in Hawthorne and the LAN server department at the Endicott Programming Laboratory. At the next step, the Hawthorne group added multimedia support. Later the project was transferred to the Client/Server Solutions Development group in the Poughkeepsie Programming Laboratory, which produced the OS/390 LAN Server product. We acknowledge the significant contributions made by Norm Bobroff, Asit Dan, Peggy Dixon, Roger Edwards, Richard Errickson, John Harter, Mark Hoffstatter, Jay Hollan, Beena Hotchandani, Don Jones, Tim Krein, Scott Marcotte, Mike Morton, Ray Rose, Joann Ruvolo, Marge Schong, Fred Schwartz, Dinkar Sitaram, Bill Tetzlaff, and Joe Williams.

Cited references and notes

1. *Protocols for X/Open PC Internetworking: SMB, Version 2*, X/Open Company, Ltd., P.O. Box 109, Penn, High Wycombe, Bucks HP108NP, United Kingdom.

2. *OS/2 LAN Server Network Administrator Reference Volume 1: Planning, Installation, and Configuration*, S10H-9680, IBM Corporation (October 1994); available from IBM branch offices.
3. *Microsoft Windows NT Resource Kit, Version 3.51, Volume 2: Windows NT Networking Guide*, Microsoft Corporation, Redmond, WA (October 1995).
4. C. C. Currie, with S. Saxon, *Novell's Guide to NetWare 4.01*, Novell Press, San Jose, CA (1993).
5. The NCP protocol is proprietary to Novell.
6. Network Working Group, Sun Microsystems Inc., Request for Comments 1094 (RFC 1094), *NFS: Network File System Protocol Specification* (March 1989); available at http://www.sunsite.auc.dk/RFC/rfc/rfc1094.html and at other Internet locations.
7. *OS/390 IBM LAN Server for MVS, Guide*, SC28-1731-01, IBM Corporation (September 1996); available from IBM branch offices.
8. A *client* is a software component that provides transparent access to server functions in a client/server environment. For instance, the SMB client translates the file requests of a client computer into the SMB protocol to be transmitted over a network to the SMB server. We use the term *client* when we talk about the specific client software implementing a client/server protocol; we use the term *client system* when we talk about the computer system that has the client function in a client/server architecture.
9. A "hard link" is a directory entry that points to the location of the file's metadata on secondary storage. This is the original UNIX method for allowing a file to appear in multiple directories with possibly multiple names. Symbolic links or "soft links" were introduced in 1983, at the University of California at Berkeley, with the BSD UNIX 4.2 operating system. A soft link specifies the target file directory entry as an absolute or relative path name and, if the target name changes, the soft link is no longer valid.
10. *ADSTAR Distributed Storage Manager for MVS, Administrator's Reference*, SH26-4040, IBM Corporation (July 1995); available from IBM branch offices.
11. K. C. Knowlton, "A Fast Storage Allocator," *Communications of the ACM* **8,** No. 10, 623–625 (October 1965).
12. K. Hwang, M. G. Kienzle, D. Sitaram, and W. H. Tetzlaff, "A Preliminary Study of Push vs Pull in Motion Video Servers," *Proceedings of the IEEE Workshop on High-Performance Communication Systems*, Williamsburg, VA (September 1993).
13. D. A. Patterson, G. Gibson, and R. Katz, "A Case for Redundant Arrays of Inexpensive Disks," *Proceedings of ACM SIGMOD*, Chicago, IL (June 1988).
14. A. Dan and D. Sitaram, "An Online Video Placement Policy Based on Bandwidth to Space Ratio," *Proceedings of ACM SIGMOD*, San Jose, CA (May 1995).
15. J. Aman, C. K. Eilert, D. Emmes, P. Yocom, and D. Dillenberger, "Adaptive Algorithms for Managing a Distributed Data Processing Workload," *IBM Systems Journal* **36,** No. 2, 242–283 (1997).
16. Information about the ATM Forum is available at http://www.atmforum.com/.

Accepted for publication February 4, 1997.

Martin G. Kienzle *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (electronic mail: kienzle@watson.ibm.com).* Dr. Kienzle received the Diplom in Informatik from the University of Karlsruhe in 1976, and the M.S. degree in computer science from the University of Toronto in 1977. In 1978, he joined the IBM Thomas J. Watson Research Center where he has been working on issues in computer architecture, operating system structure, and performance. In 1992 he received the Ph.D. degree from the Department of Electrical and Computer Engineering at the University of Massachusetts at Amherst. Dr. Kienzle, a Senior Technical Staff Member, manages the Parallel Multimedia Servers group, which analyzes and develops video server software. Examples of this work are the file system, the resource manager, and the CLAW component of the OS/390 LAN Server, and the control server that served as part of the base for the AIX® video server products. Dr. Kienzle's research interests are in distributed multimedia servers and architectures, and algorithms related to multimedia data storage and delivery. Dr. Kienzle is a senior member of the IEEE (Institute of Electrical and Electronics Engineers) Computer Society and a member of ACM.

Robert R. Berbec *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (electronic mail: berbec@watson.ibm.com).* Mr. Berbec earned his B.S. degree from Harvey Mudd College in mathematics in 1966 and his M.S. degree from Stanford University in statistics in 1968. He has been employed by IBM since 1968, where he is a senior programmer. His technical interests are in large systems data management.

Gerald P. Bozman *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (electronic mail: bozman@watson.ibm.com).* Mr. Bozman is a Senior Technical Staff Member with a special interest in operating systems and file systems. He has worked on System Managed Storage for MVS/DFP; the Parallel Processing Compute Server, a prototype 390 multicomputer; the NetWare Gateway to OS/390 LAN Server; and has made performance contributions to products in the areas of caching, file systems, and storage management.

Catherine K. Eilert *IBM S/390 Division, 522 South Road, Poughkeepsie, New York 12601 (electronic mail: eilert@vnet.ibm.com).* Mrs. Eilert is a senior software engineer in the S/390 Performance Design department. She was the team leader for the workload management algorithms project and received an IBM Outstanding Technical Achievement Award for that work. Mrs. Eilert holds four issued patents related to performance management algorithms. She received a B.S. degree in industrial engineering and an M.S. degree in computer science, both from the Illinois Institute of Technology in Chicago. Her current interests are performance management algorithms and designing software for performance and scale.

Marc Eshel *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (electronic mail: eshel@watson.ibm.com).* Mr. Eshel received the B.Sc. in computer science from City College of New York and the M.Sc. in computer science from Pace University in White Plains, New York. Mr. Eshel joined IBM Endicott in 1982, where he worked on VM/CMS (virtual machine/conversational monitor system). In 1985 he joined IBM Research and is now a senior programmer. Most of his work is in the area of operating systems, in particular on file systems. On recent projects he contributed to the products Multimedia File Serving for OS/390 and IBM Multimedia Server for AIX.

Raymond Mansell *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (electronic mail: mansell@watson.ibm.com).* Mr. Mansell joined the IBM UK Laboratories in 1974 after receiving a bachelor's degree in electronic engineering from the University of Bath. He transferred to the Research Division in 1990 and has specialized in work on high-performance file systems.

Reprint Order No. G321-5649.