Multimedia Databases and Servers

Warren Sterling
Felipe Cariño
Catherine Boss

Database technology is evolving to accommodate complex multimedia objects—data types that present unique challenges in the storage and movement of large time-dependent objects. The manipulation and analysis of these large and semantically rich objects require techniques far different from those used for alphanumeric databases. This paper presents multimedia server-based architectures based on massively parallel processor systems and an object-relational database management system. The transition from client-centric multimedia applications to a client/server environment is examined, and several applications are discussed that illustrate the use of multimedia database servers based on the object-relational model. Finally, this paper reviews the architecture of two scaleable multimedia servers that can support these applications: MoonBase, a database server with full content analysis capability for multimedia objects, and the audio/video server, a system-based file server that can operate within the MoonBase architecture.

Introduction

Multimedia databases and servers are keys in the convergence of communications and computers to create innovative information processing solutions. Businesses are beginning to understand that a multimedia capability allows them to store and convey information to their customers and employees more effectively, thereby increasing their competitive advantage. As businesses incorporate multimedia *objects*—such as full-motion video, audio, pictures, complex documents, graphics, and animation—into their "mission-critical" data, database technology must expand to accommodate these complex data types and provide users convenient access to their rich semantics.

Conventional database servers store, retrieve, transform, and analyze information. Multimedia database servers must implement these same operations, not just for traditional alphanumeric data, but for complex data types, which include multimedia objects. These servers must be scaleable—that is, incrementally expandable in terms of storage capability, processing power, and network connections as the size of the customer application grows.

Customers increasingly are implementing enterprise-wide applications—that is, applications serving the entire geographic and functional areas of a business, including global businesses, and storing data representing the entire business. The size of multimedia objects, such as video data, and the processing power required to manipulate and analyze such objects dictate that enterprise-wide servers potentially will require petabytes (10^{15} bytes) of storage and multi-BIPs (billions of instructions per second) of processing power.

Additionally, these systems must support the sophisticated multimedia communication services required for visual real-time collaboration and other applications where time-dependent multimedia data stored on a server must be delivered to a user with a guaranteed quality of service. A *massively parallel processor (MPP)* system—hundreds of physi-

Panel 1. Abbreviations, Acronyms, and Terms

ADT — abstract data type
ANSI — American National Standards Institute
ATM — asynchronous transfer mode
AVS — audio/video server
BeSS — Bell Laboratories Storage System
CPU — central processing unit
FDDI — fiber distributed data interface
GDD — global data dictionary
GUI — graphical user interface
H.221— industry-standard protocol for multiplexing conferencing audio, video, and data (equivalent to Px64)
H.320 — suite of industry standard protocols that includes H.221 and data conferencing
I/O — input/output
intelligent agent — software that performs tasks, searches for data, compiles and presents information, and makes or aids in decisions on behalf of a user
ISDN — integrated services digital network
isochronous transmission — time-dependent data, such as a video data stream arriving at its destination at the constant rate required for processing or display
JIT— just in time
LAN — local area network
MCU— multipoint control unit for switching 64 kb/s ISDN lines
MPEG-1— Motion Picture Experts Group, video compression standard
MPP— massively parallel processor
MRI — magnetic resonance imaging
OCR— optical character recognition
ODE — object database and environment
OID— object identifier
OLCP — online complex processing
OLTP— online transaction processing
OM — object manager
OSC — object server connectivity
PC — personal computer
PCS — personal conferencing specification
petabyte — 10^{15} bytes
predicate — a user-specified condition that must be satisfied to retrieve a tuple.
Q.931 — digital subscriber signaling system No. 1 for ISDN users
RAID — redundant arrays of inexpensive disks
RASR — reliability, availability, serviceability, recoverability
RDBMS — relational database management system
SCSI — small computer systems interface
SQL3 — proposed definition of structured query language with object support
TCP/IP — transmission control protocol/Internet protocol
TPC — Transaction Processing Performance Council. TPC-A, TPC-B are OLTP benchmarks; TPC-C is an OLCP benchmark; TPC-D is a benchmark for complex decision support processing
tuples — rows, or records of a relational database table
UDF — user-defined function
VCR — video cassette recorder
WAN — wide area network

cally and functionally independent processors interconnected to form a single system—is well suited to satisfy these storage, power, and scaleability requirements.

This paper examines the role of the relational database model as a multimedia server for three business applications—business training, information services, and user collaboration—and the server architectures which support these applications.

Then, the AT&T Global Information Systems (AT&T-GIS) multimedia database server, MoonBase, and its associated object managers, Prospector and the audio/visual server (AVS), are described. Designed to exploit MPP systems, these servers support each of the illustrated application server architectures. These server designs are based directly on our experience with the high-end AT&T Teradata® Database System, which has successfully handled multi-terabyte (10^{12} bytes) databases with complex query performance that is

unmatched by any other database vendor.

Client-Centric Versus Client/Server

Most multimedia applications today can be classified as client-centric in that the storage, manipulation, and presentation of data all take place on a single client platform, such as CD-ROM-based encyclopedias or games, or on multiple peer-client platforms, such as the AT&T Vistium™ Personal Video System.[1] Client-centric applications, however, give rise to "islands" of multimedia information throughout an enterprise, and offer little capability for sharing and great potential for duplication and inconsistency.

The move to enterprise-wide applications requires a migration to servers because of the following:

- Size: The massive storage requirements for multimedia objects preclude the use of storage systems on clients.
- Data consistency, sharing, accounting: Data can be maintained with consistency and shared more easily when stored on a server. Asset management, such as royalties associated with the use of copyrighted multimedia objects, can be more effectively handled on a server.
- Security: Controlled enterprise-wide access to multimedia objects, including create, update, and delete capabilities, can be more effectively controlled in a server environment.
- Reliability, availability, serviceability, recoverability (RASR): Enterprise-wide applications often require availability 24 hours a day, 7 days a week. This is virtually impossible to maintain in a client-centric environment; however, RASR can be designed into a mission-critical server.

Database Technology

There are many database management systems, such as hierarchical, network, relational,[2] and object-oriented[3] systems, that are used in a ubiquitous range of business and scientific applications. Database systems provide the facilities and mechanisms to handle concurrency control, transaction management, integrity constraints, system administration, and other functions. Early database applications did not use database management systems for performance reasons, and chose to store the database in files. These applications, known as *legacy systems*, are difficult to enhance and maintain.

History seems to be repeating itself, since some developers of multimedia database applications are still using files because current commercial database systems were not designed to handle very large multimedia databases. The AT&T-GIS MoonBase multimedia database server, however, was designed to overcome the performance problems that commercial multimedia database systems can't handle.

The two main database technologies that are of interest to multimedia users are *relational* and *object-oriented*. An analysis of their technical and market benefits is beyond the scope of this paper, and architectural discussions here are limited to file system-based servers and object-relational database servers. The object-relational model is an evolution of the relational model to incorporate object support. Multimedia "objects" are included in the general class of "objects." A comprehensive discussion of extended and object-oriented databases can be found in Cattell et al.[3] The terms object and object-oriented have diverse meanings and even books[4] on the subject avoid defining the terms. A MoonBase SQL3-like object can be thought of as a C++[4] class definition that defines complex abstract data types (ADTs) and user-defined functions (UDFs) to manipulate the class (object).

Business Applications

Business applications where multimedia data add significant value can be organized into three general categories:

- Business training, where timely delivery, cost-effective storage and delivery, and client-based tuning of material are valuable;
- Information services, where full-motion video, high-quality audio and images, and other complex data types bring new forms of information to businesses; and
- User collaboration, where users can more effectively communicate and share information with notes, annotations, and voice-overs in an interactive data and message application.

These three architectures, illustrated in Figures 1, 2, and 3, have several common elements:

- *Multimedia-enabled clients*, such as the AT&T Vistium Personal Video System or other PC-based systems, communicating over a local area network (LAN) or wide area network (WAN). The WAN can be the telephone switching network, as in the user collaboration applica-

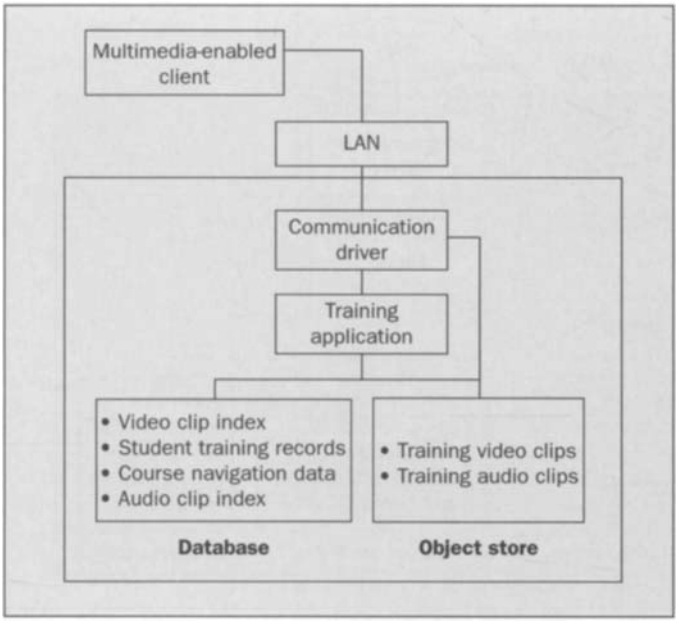


Figure 1. In this high-level view of a business training multimedia application, the application database holds all alphanumeric information required to identify and access specific multimedia training objects, and it can hold any client-specific information. The object store holds the multimedia objects created by the education experts. The training application accesses the database via the LAN to identify the proper video or audio clip to present to the student. It then accesses the object store to set up a data stream link between the object store and the communications driver to deliver the selected clip to the client.

tion that is discussed below.

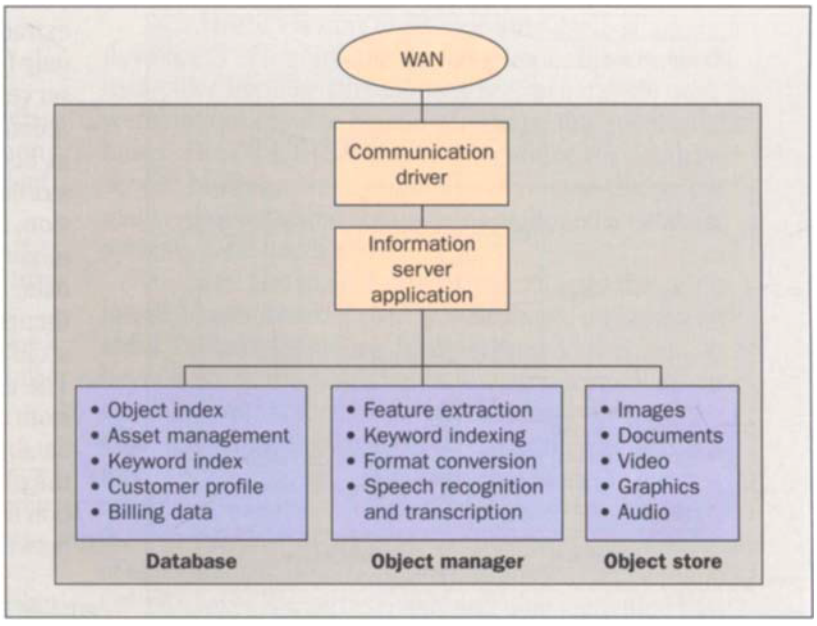
- A *communication driver*, which understands the interfaces to the network and to the application software running on the server, can drive time-sensitive data streams, such as video and audio, onto the network or accept them from the network, and generally orchestrates the operation of the server.
- An *object store*, which handles the physical storage of the multimedia objects. This consists of the physical storage media, such as magnetic disks, optical disks, and other technologies, in a hierarchical storage management system, and the file system designed to handle them.
- An *object manager*, which can execute the stored multimedia objects using such functions as image feature extraction and keyword index generation (required only for the information server and the collaboration server).
- A *database*, which stores what is called *metadata*, that is, both user and system information related to each stored object. For purposes of this application discussion, no specific database model is assumed. In some models, such as object-oriented databases, the metadata, object manager functions, and the stored objects themselves are much more tightly linked than this architecture implies.
- The *application software*, which interprets commands from the clients and executes them by querying the database for information, initiating functions stored in the object manager, and establishing the communication linkages required to transfer multimedia objects between clients and the object store.

Business Training. Business training terminology applies to a wide variety of applications, from the concept of just-in-time (JIT) training to the use of remote, or offsite classrooms. The course information, written by education and subject matter experts, is stored in a central location on a server and delivered to the desktop. Traditional lecture- and textbook-based training applications can be augmented with video and audio, while both textbook- and CD-ROM-based training can be augmented by using a central location for course information, for course navigation based on a student's abilities, and for administrative utilities, such as course and student tracking and grading.

The object store in Figure 1 holds the multimedia objects created by the education experts. The database holds all the alphanumeric information required to identify and access specific multimedia objects. It also can hold all student-specific information. The training application accesses the database to identify the proper video or audio clip to present to the student. It then accesses the object store to set up a data stream link between the object store and the communications driver to deliver the selected clip to the student, assumed here to be using a multimedia workstation. In this case, there are no functions that must be executed against the stored objects, so no object manager is required.

Information Services. For the consumer, information services can include customized CNN*-style television news information delivered to the desktop, shopping

Figure 2. This illustration shows a general architecture for an information server. The multimedia objects stored on the server are quite general in nature, given the wide ranging domains of information servers. The object manager manipulates and analyzes the multimedia objects in the object store, and contains application-specific functions to do so. The database also contains information to identify and locate objects (the object index) and information concerning usage of the server (customer profiles and billing data).



via either cable TV services or kiosks in stores, and continually updated weather forecasts integrated with airline flight information. Existing examples of information services, such as *America Online** and AT&T's *Interchange Computing,*® are delivered via WAN-based services, including the Internet and the World Wide Web. For example, AT&T has implemented a prototype real estate application using the Intuity™/CONVERSANT® Multimedia Transaction Processing System to allow prospective home buyers to view pictures and floor plans of homes for sale.

For business, these services cover many applications in such areas as general document and image management, transaction authorization and verification, customer relationships, customer service, merchandising, manufacturing, and equipment maintenance.

For brevity, we examine in more detail only the first of these—general document and image management—which involves eliminating the hard storage of paper, images, and audio transcripts. Examples of this application are:

- ImageWeb from AT&T Business Communications Services, where complex documents are stored centrally, electronically delivered to customer end-points upon request, and then printed; and
- Scaleable Image Item Processing System (SIIPS) from AT&T-GIS, where images of checks and other financial instruments are captured, automatically processed using optical character recognition (OCR) to extract information from the check, and retrieved for display by check processing operators.

In the insurance industry, great benefit can be derived by storing all information associated with insurance policies and claims in a multimedia database. This would include, for example, alphanumeric policy data, photos and videos of insured objects, scanned images and audio recordings of customer correspondence, photos of damages, and recorded witness statements. An agent or claims adjuster responding to a customer could access—from one souce—all the relevant documentation on a claim. Similar scenarios exist for medical records and bank loan documentation.

The United States Postal Service has proposed creating a nationwide network of kiosks, called *The Government Connection*, to provide the American public with the means to transact business with local, state, and federal agencies and service providers. Multimedia data must be supported for this application, such as video instructions and the capture of the user's picture for license applications.

Figure 2 shows a general architecture for an

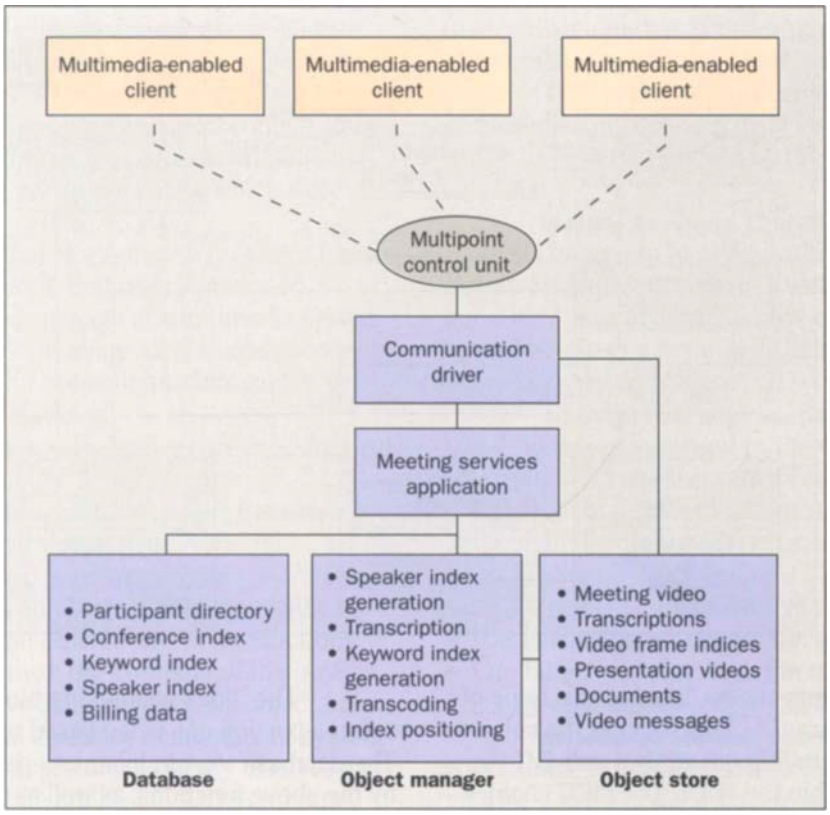


Figure 3. This illustration shows a general architecture for a collaboration server for videoconferencing. Clients access the collaboration server through a multipoint control unit (MCU), which is maintained within the wide area network. The MCU enables videoconferencing among the parties, as well as gives them access to the server.

information server. The multimedia objects stored on the server are quite general in nature, given the wide-ranging domains of information servers. Note the addition of an object manager, which manipulates and analyzes the multimedia objects in the object store, and contains application-specific functions to do so.

Typical functions are listed in the figure. For example, the object manager may have a feature extraction function that can analyze video tapes taken in a department store to better understand shopping patterns. It also could contain functions to scan complex documents and generate keyword indices, and place the results in the database. The database also could contain information to identify and locate objects—the object index, as well as information concerning the use of the server—customer profiles and billing data. Asset management could include the important function of tracking the use of copyrighted multimedia data for royalty payments. These are just a few examples of the many functions that can be implemented on an information server.

User Collaboration. Collaboration refers to the ability of people to communicate with others and work on single tasks—no matter what their location and their time zones—as easily as they do in person. It involves the ability to simultaneously view, annotate, and update documents that have been previously stored. When individuals cannot link up at the same time, they should be able to create and view stored video messages, accompanied by other stored multimedia information, such as documents, images, and video.

Examples of collaboration services include proposal reviews, advertising reviews, medical consultations, real-time videoconferencing, and video messages. Collaboration servers can provide enterprise-wide access to all of the multimedia elements of remote interactive work, including recordings of videoconferences.

Figure 3 shows a general architecture for a collaboration server for videoconferencing. Examples of

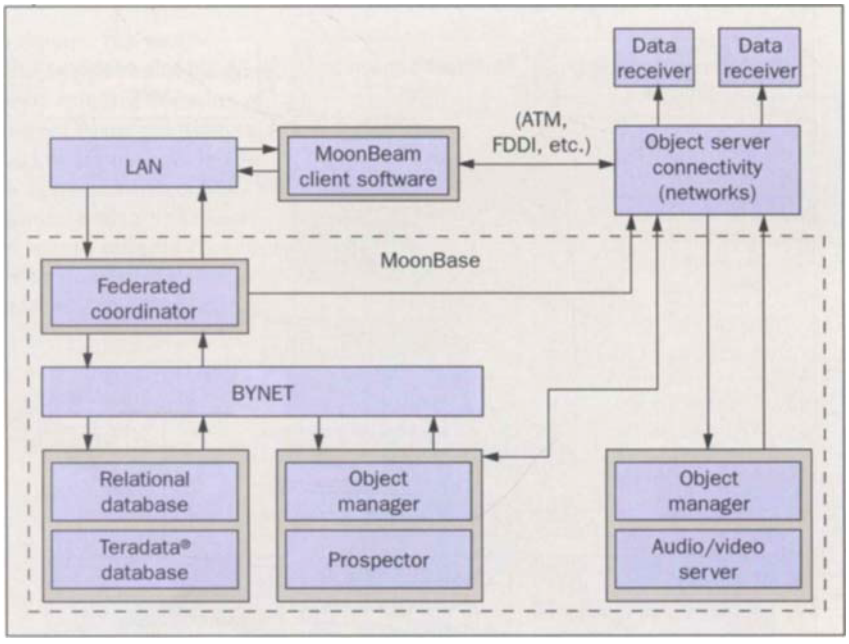


Figure 4. MoonBase is a multimedia object-relational database being developed for new emerging multimedia applications. MoonBase contains an extensible federated relational database coordinator that non-intrusively adds multimedia capabilities to existing relational database management systems. The coordinator also logically unifies two database components—the relational database and the object manager.

multimedia-enabled clients can be found in this issue of the *AT&T Technical Journal*.[5] Clients access the collaboration server through a multipoint control unit (MCU), which is maintained within the WAN. The MCU enables videoconferencing among the parties, as well as giving them access to the server. The object store holds video recordings of meetings and presentations, transcriptions of the audio portion of recordings, documents to be shared during videoconferences, video messages, and individual video frames that can be used to display a visual index for specific meetings and presentation videos.

As in the information server, the object manager contains functions that operate on the stored multimedia objects. For example, the transcription function performs a voice-to-text conversion on the audio portion of video recordings of meetings, and stores the transcript as a multimedia object in the object store. The index-generation function analyzes meeting videos to detect when the speaker changes, and generates a video frame index to mark it. The keyword index-generation function analyzes transcripts to generate indices that are stored in the database. The transcoding function converts video stored in one format, such as Px64, to another format, such as personal conferencing specifications (PCS), before the data is streamed to a client.

The index positioning function allows a client to select a portion of a video based on the stored indices. The database stores alphanumeric information generated by the above functions, as well as other administrative data required to operate the videoconferencing system, such as directories and billing data.

In the remainder of this paper we examine two specific architectures—MoonBase and the audio/visual server—for multimedia databases and servers.

The MoonBase Architecture

The MoonBase server (see Figure 4) supports a multimedia object-relational database that is being developed by AT&T-GIS for new emerging multimedia applications. MoonBase contains an extensible[6] federated[7,8] relational database coordinator that non-intrusively adds multimedia capabilities to existing relational database management systems. (Federated databases and servers can act on their own or be coordinated to act in parallel, yet look to the user like just one system.)

The MoonBase coordinator logically unifies[9] two database components: the relational database and the object manager. A relational database system, like the Teradata Database,[10,11] is used to store user table attribute values, metadata, a global data dictionary (GDD), and

object identifiers (OIDs). OIDs are links in the relational database tables that point to objects stored in object managers. The object managers are used to store and retrieve objects, including retrievals based on object content analysis. For the purposes of this architectural discussion, the object managers include the object stores, which were shown separately earlier in this paper.

MoonBase is the first attempt to provide a framework for a complete solution that uses a federated approach to integrate and manage all multimedia system components and technology, since large time-dependent multimedia data types and projection issues cannot be handled by traditional databases.

Now we will discuss the salient areas addressed by MoonBase.

MoonBase Query Language and Object Definition. This facility is based on the American National Standards Institute (ANSI) structured query language (SQL3).[12] SQL3 provides an abstract data type and user-defined function to define objects and write functions that manipulate and analyze the contents of the objects. MoonBase adds selective SQL3 multimedia object functionality to any relational database management system (RDBMS). MoonBase will provide a library of ADTs and associated functions, while additional ADTs and functions can be developed by professional services organizations or the customers themselves.

Two-Pass Approach. This two-pass approach is used to store or retrieve *tuples,* the term for rows, or records, stored in a relational database. If a user in the first pass retrieves multiple rows containing several large multimedia objects, like images and video, the retrieved tuples will contain alphanumeric data and corresponding OIDs, which represent specific multimedia objects. In the second pass, the user or an intelligent agent[13] selects the objects that are to be retrieved, using the corresponding OIDs to address them. An intelligent agent is software which performs tasks, searches for data, compiles and presents information, and makes or aids in decisions on behalf of a user.

Object Server Connectivity (OSC). The OSC provides the logical and physical capabilities to use multiple physical network connections. Typically a LAN connection is used to send queries to the MoonBase coordinator. The first-pass alphanumeric results are returned to the client over the same LAN. In the second pass, the OSC is used to establish a higher bandwidth connection between the object manager and the requesting client. The significance of OSC is that MoonBase manages, coordinates, and handles network security issues that otherwise would have to be handled by the application programs or users.

Multiple Receivers. This concept allows the query requests to be sent to other multiple data receivers (see Figure 4). A query originates with the MoonBase client, but multimedia objects selected by the query can be broadcast directly to multiple data receivers that did not initiate any query activity.

Graphical User Interface (GUI). The GUI incorporates the two-pass approach, the OSC multiple-network concept, the multiple receivers concept, and other GUI[14] visual ergonomic issues that pertain to the nature of large, multiple multimedia objects. MoonBase is based on a client-server architecture. The MoonBase client GUI software runs on any multimedia computer that:

- Has isochronous networking; or
- Can buffer and play back the multimedia data objects, such as full-motion video, still images, and audio.

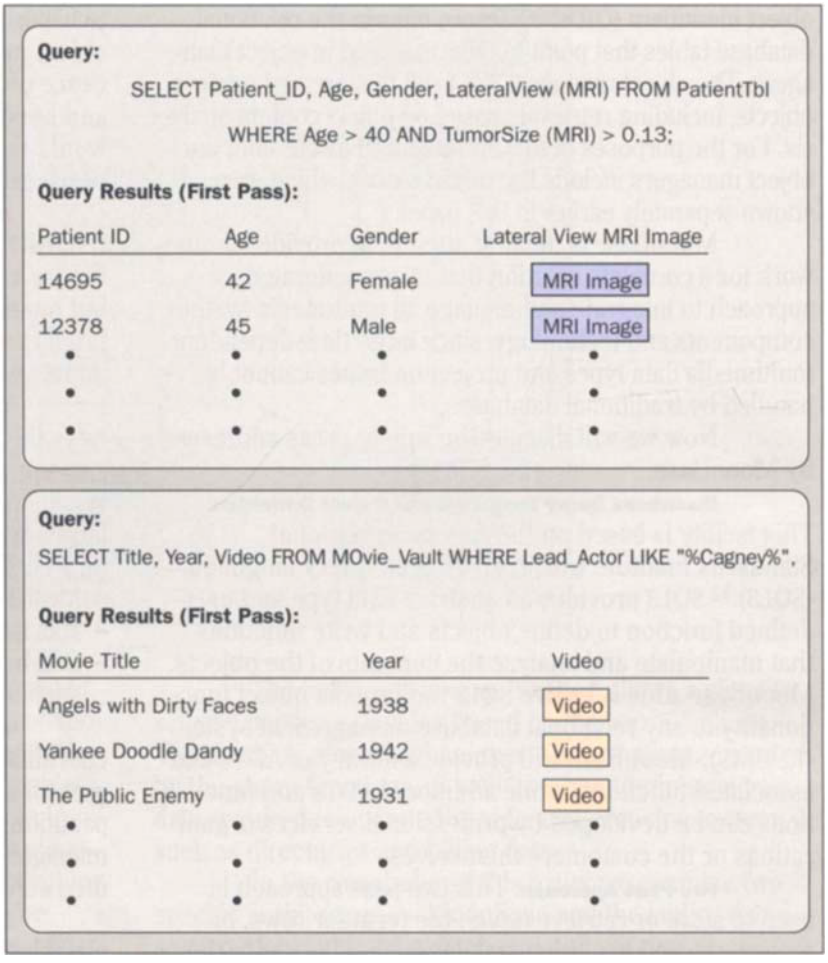
MoonBase Federated Coordinator. The federated coordinator parallelizes and develops execution strategies for a *share-nothing architecture*[15] that is "vertically" partitioned between the relational database and object managers. Share-nothing data are tables evenly distributed across all processors making up an MPP.

Multiple Object Managers. MoonBase uses multiple object managers to store and retrieve multimedia data. Prospector, MoonBase's principal object manager, is a general purpose high-performance multimedia object manager that provides fast storage and retrieval of a large amount of multimedia data. It also allows the parallel execution on massively parallel platforms of user-defined content analysis algorithms on multimedia objects.

Prospector, which exploits parallelism by running on multiple processors in an MPP system, is built upon AT&T Bell Laboratories' Storage System (BeSS).[16] Video and audio servers can be third party object managers, such as the AVS server, that have special bandwidth and throughput requirements. The RDBMS component can be any relational database system.

The two examples in Figure 5 are based on applications previously explained. The first query demonstrates the capability of a content-based *query*

Figure 5. The medical query in Figure 5 (top) shows the abstract data type (ADT) and user-defined function (UDF) capabilities of structured query language (SQL3), as discussed in the text. To view an MRI image, the user would select an icon and the image would be transferred to the MoonBeam client for display. The movie query in Figure 5 (bottom) also returns to the user alphanumeric values and icons representing multimedia objects, in this case the video movies.



predicate, a user-specified condition that must be satisfied to retrieve a tuple. The second query illustrates the retrieval of video movies using an alphanumeric query predicate. Both queries use the two-pass approach and OSC transport connections that are designed to retrieve large objects and time-dependent objects.

The medical query in Figure 5 (top) shows the ADT and UDF function capabilities of SQL3. In this example of a medical application, the user has defined magnetic resonance imaging, or MRI, as an ADT and used it as a column of a table. This query also shows two UDF functions that operate on the MRI ADT attribute column: LateralView and TumorSize. The LateralView function operates on and understands the MRI ADT properties. In this case, TumorSize returns a number that represents that size of the patient's tumor. MoonBase then applies the LateralView projection UDF to all the MRI attributes that matched. The query results show the first pass answer set. The rectangular icons represent the actual MRI image.

To view an MRI image, the user would select an icon and the image would be transferred to the MoonBeam client for display. This image transfer, and all other image transfers based on the first pass answer set, represent the second pass of the query results.

The movie query in Figure 5 (bottom) also returns alphanumeric values and icons representing multimedia objects, in this case video movies. Again, the user selects an icon to view the movie, which would be transferred isochronously and viewed under MoonBeam's virtual video cassette recorder (VCR) control. MoonBeam provides a user on-screen VCR-like capabilities, such as play, rewind, and fast forward.

MoonBase integrates technologies, such as relational databases, file systems, video servers, and networks that have developed benchmarks to help evaluate and compare systems. For relational databases, there are a set of benchmarks (TPC-A, -B, -C, -D) that are used to evaluate relational systems for on-line transaction processing (OLTP), on-line complex processing (OLCP), and complex decision support applications.

There also are benchmarks proposed for object-

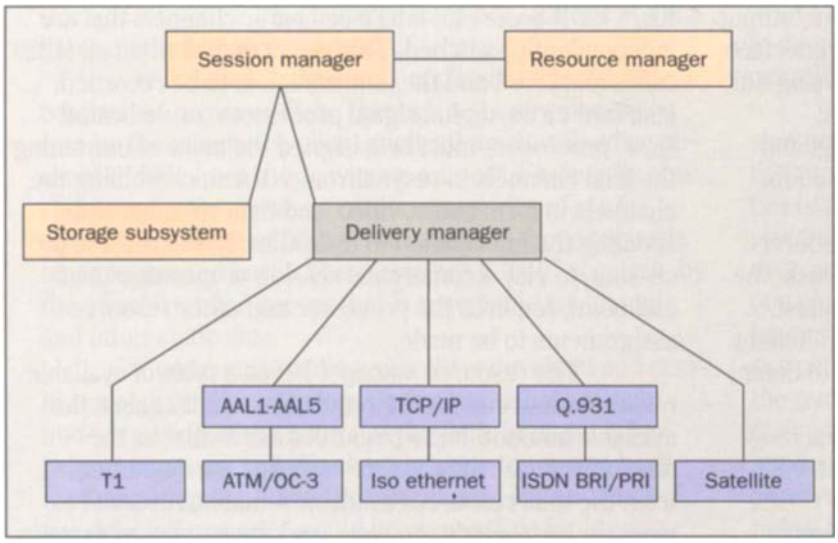


Figure 6. This illustration shows the internal architecture of an audio/video server (AVS), which can operate as a MoonBase object manager and is optimized for continuous media storage and retrieval. When the AVS is asked to deliver continuous media to the client, it is given the content object identification, content delivery characteristics, and the destination address by MoonBase.

oriented database systems,[17] although these benchmarks have not evolved to include SQL3 object-relational benchmarking. Therefore, new benchmarks are needed for systems like MoonBase that perform complex content analysis operations on objects, and transfer from the object store to a client large time-dependent objects, such as video, and time-independent objects, such as images.

Audio/Video Server

This section discusses the internal architecture of an AVS, which can operate as a MoonBase object manager and is optimized for continuous media (time-dependent data) storage and retrieval. It is assumed that MoonBase places content in the AVS, processes initial requests from clients, and logs billing and usage information. When the AVS is asked to deliver continuous media to the client, it is given the content object identification, content delivery characteristics, and the destination address by MoonBase.

The AVS components and their linkages are shown in Figure 6, and each of the components is described in more detail below.

Session Manager. The session manager is the "traffic cop" of the delivery. Initial incoming requests are either denied or granted by the session manager, based on the availability of resources. (Media control commands, or upstream signaling, that occur during the session are handled by the delivery manager as described in a subsequent section.) The session manager searches and updates the session database on a per-session basis. The session database contains one record per session, including:

- User information, such as a client's address;
- Connect time, updated periodically by the session manager;
- Services required during the session;
- Content delivered in the session; and
- Active flag that describes whether the session is active, versus a complete flag that describes whether the session is complete;

The session database is used to make billing decisions and do user profiling.

The session manager is responsible for restarting failed sessions, as notified by the delivery manager when the connection to the endpoint is lost. The session manager also interfaces with the storage manager when new content comes in for placement or when content is deleted.

Resource Manager. The resource manager is responsible for tracking, allocating, freeing, and analyzing the use of server and network resources. The resource manager updates and searches the resource database for information.

The resource database, the repository for information on available resources in the system, contains:

- Currently available resources such as the input/output (I/O) bus bandwidth, small computer system interface (SCSI) host adapter bandwidth, central processing unit (CPU) cycles, buffer space, and network ports;
- Resources that each session is currently using; and
- Information on denied requests and the reason for the denial.

When the session manager asks the resource manager if resources are available for a certain task, the task characteristics must be passed with the request. Task characteristics must at a minimum include content bandwidth requirements and processing needs to determine the system resources required.

There are two tasks with disparate characteristics that are typical of multimedia workloads that will be modeled and must be understood.

Streaming MPEG. The first task is streaming Motion Picture Experts Group (MPEG) video at 1.5 Mb/s over an asynchronous transfer mode (ATM) network to a client for playback. Resources associated with this activity include:

- Network bandwidth available;
- Disk latency given the current disk workload, that is, the time it takes for the disk head and disk to conclude one operation and move into position to provide information for the next operation;
- Memory bandwidth if the data transfer from disk to network is not peer-to-peer;
- I/O bus bandwidth; and
- The availability of the on-board processor to generate the ATM cell header and package data bits into cells.

Note that much of the associated I/O activity depends on balancing the amount of on-board buffer memory—and thus I/O block transfer size and frequency of data transfers—with system-level activity generated by each data transfer.

Recording H.320 Data. The second task is recording H.320 conferencing or messaging data from an integrated services digital network (ISDN) or ATM network to a data file and playing it back through the network. Again, disk resources and I/O bus bandwidth must be monitored, but given the relatively low bandwidth requirements of conferencing and messaging today, typically 64 Kb/s at the low-end and 384 Kb/s at the high-end, processing power becomes much more of an issue.

H.320 is delivered typically over one to six 64-Kb/s ISDN basic rate interface bearer channels that are independently switched. This requires that when an H.320 call is answered and the information is to be recorded, standard CPUs, digital signal processors, or dedicated H.320 processors must be assigned the tasks of unframing the data channels to resynchronize them; combining the channels to form audio, video, and data streams; and spooling the information to disk. Clearly, the reverse processing, to play a conference session or message to an end-point, requires the processor and other resource assignments to be made.

The resource manager keeps a table of available resources and checks the required amount against the available amount. If the resources are available, the amounts of each appropriate resource are decremented from the total resources available—that is, reserved to meet the customer's request—and the session manager's query is accepted. If the resources are not available, the session manager's query is rejected. On completion of delivery, the resource manager returns the resources to the pool, based on the linkage between the reserved resources and the session.

The intent is to build a model of resource use such that certain task types have associated with them x amount of processing capabilities, disk I/O, memory, and network resources. Those requirements can roughly be estimated based on observation and tuned by further characterizing the task.

One can also better understand each individual customer's needs based on the customer's typical workload. As the knowledge base is built, the appropriate configuration for a customer can be predicted based on the customer's workload. One can also determine, when a customer is dissatisfied with the server's performance, where the bottleneck is in the system and recommend a specific upgrade.

Note that these tools and the concept of resource reservation are applicable to all customers' workloads, including OLTP, file server, and network management.

Delivery Manager. The delivery manager is responsible for orchestrating the movement of data from the storage subsystem to the network connection. The delivery manager abstracts the network interface to the upper layers of software. There are a few alternatives, including "pull" and "push" scenarios, for the role of the delivery manager, dependent on the level of intelligence in the

storage subsystem, the network interface, and the application's method of accessing data.

"Pull" scenario. As noted, applications today access continuous media as files. Through a set of well-defined remote file interfaces, client applications download a subset of the file from the server on request. Essentially, the client application issues a continuous series of *read* requests for data, and the server satisfies those requests on an as-needed basis. This describes a "pull" scenario, by the client from the server that is acceptable for textual and other static data.

Continuous media types allow the client and server to cooperate for optimum performance based on an understanding of the nature of the data, in that it is logically contiguous, and the client is likely to want to have the "next" segment delivered eventually. In this situation, the delivery manager would be responsible for issuing *read-ahead* requests, buffering the read-ahead data on the SCSI card or the network interface card, and issuing the delivery of the buffered data on request.

"Push" scenario. The second scenario in continuous media delivery is a "push" from the server to the client, where the initial job requests come in from the client, and the server takes responsibility for delivering the data at the appropriate rate to the desktop. The client and server negotiate buffer and playback capabilities to determine appropriate rates. In this case, the delivery manager must have the intelligence to fill buffers in real-time on the SCSI or network card, based on an understanding of the continuous *media playlist*. A playlist is a linked list of buffer-sized chunks of the content and their location on disk. The delivery manager must have enough real-time capabilities to recognize when buffers are empty and fill them in such a way that underflow or overflow conditions do not occur.

The delivery manager also intercepts continuous interactive media control commands, such as fast-forward, rewind, and pause. The delivery manager determines how to move forward and backward in a playlist of content buffers or move forward and backward a number of bytes, corresponding to x seconds in a file.

Storage Management

Storage management includes placing new content on disk, retrieving that data for delivery, and removing content. Continuous media can be optimally placed to minimize *disk seek latency*—the time it takes for the disk head to move radially and position itself over the disk—to maximize the simultaneous numbers of viewers, or can, alternately, be placed to provide less-costly storage.

New content is placed via a request from the session manager. Placement algorithms are determined by the application's requirements. At least two characteristics of the application come into play when determining placement algorithms, *striping* or *dynamically cloning*. A disk today can support roughly five to seven simultaneous playbacks of an MPEG 1 video stream; the issue is latency and seek time. The choice is based on cost versus performance. The application can choose to increase the number of simultaneous viewers via striping across disks, thus reducing the activity on a single disk, which implies hardware or software redundant arrays of inexpensive disks (RAID) or a proprietary solution. The alternative is to dynamically clone the content across host adapters or disks, thus increasing the amount of storage required.

Deleting content is initiated by the session manager, which issues a series of file removal commands to the storage manager based on the delete request from MoonBase. Note that the master copy and its clones must be deleted simultaneously.

For content playback, the delivery manager issues a series of job requests to the storage manager for buffer-sized retrievals from disk and placement in network card buffers. Buffers are then played to the desktop by the network card.

Network Subsystem. The network cards and drivers are responsible for applying appropriate protocols to the buffered data. This may include packetization, in the case of TCP/IP deliveries, or header generation and organization into cells, as is the case with ATM. These interface cards and drivers must have enough intelligence to meet bit rate requirements and generate interrupts to the delivery manager when buffers need to be filled. They must also detect failed connections or other delivery problems and notify the delivery manager, which notifies the session manager when problems have occurred.

Upstream signaling must generate high enough priority interrupts that requests for action are met to the end-user's satisfaction. Network software communicates with the delivery manager to handle media control commands.

Summary

We have described the migration to multimedia databases and servers, discussed general application categories and server architectures, and examined the architectures of a multimedia messaging server and a multimedia database server. Clearly, the sheer size and time-dependent nature of many multimedia data types put stringent demands on all key aspects of server design: network communications, storage capability, processing power, and scaleability.

At the conceptual level, database features and functions apply equally well to alphanumeric or multimedia data types, but multimedia data types introduce new semantic issues and, more importantly, require new strategies for manipulating and moving extremely large objects. The multimedia databases and servers described here will allow businesses to store, retrieve, manipulate, and analyze collections of multimedia data types to solve a wide range of new business applications as businesses modernize their processes.

(Manuscript approved July 1995)

***Trademarks**

CNN is a registered trademark of Turner Broadcasting Corporation.
America Online is a registered trademark of America Online Inc.
UNIX is a registered trademark of Novell in the United States and other countries, licensed exclusively through X/Open Company Limited.

References

1. AT&T Vistium User Guide, ST-2129-76, AT&T Global Information Solutions, Dayton, Ohio, June 1994.
2. E. F. Codd, "A Relational Model for Large Shared Data Banks," *Communications of the Association for Computing Machinery (CACM)*, Vol. 13. No. 6, June 1970, pp. 377-387.
3. R. G. G. Cattell, *Object Data Management–Object-Oriented and Extended Relational Database Systems*, Addison-Wesley, Reading, Massachusetts, 1991.
4. B. Stroustrup, *The C++ Programming Language*, Addison-Wesley, Reading, Massachusetts, 1986.
5. P. Crouch, B. Schwartz, J. Rodriguez, "Screen-Based Multimedia Telephony," *AT&T Technical Journal*, Vol. 74, No. 5, September/October 1995, pp. 78-91.
6. E. F. Codd, "Extending the Relational Model to Capture More Meaning," *Association for Computimg Machinery Transactions on Database Systems*, Vol. 14, No. 4, December 1979, pp. 397-434.
7. F. Cariño, "HETERO - Heterogeneous DBMS Frontend," *Proceedings of the International Federation Information Processing Technical Committee (IFIP TC) 8/WG 8.4 Working Conference on Methods and Tools for Office Systems*, October 1986, pp. 159-172.
8. A. P. Sheth and J. A. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases," *Association for Computing Machinery (ACM) Computing Surveys*, Vol. 22, No. 3, September 1990, pp. 183-236.
9. W. W. Chu and I. T. Ieong, "A Transaction-based Approach to Vertical Partitioning for Relational Database Systems," IEEE *Transactions on Software Engineering*, Vol. 19, No. 8, August 1993, pp. 804-812.
10. F. Cariño and P. Kostamaa, "Exegesis of DBC/1012 and P-90," *Proceedings of the 4th International Parallel Architectures and Languages Europe (PARLE `92)*, Springer-Verlag, pp. 877-892.
11. F. Cariño, W. Sterling and P. Kostamaa, "Industrial Database Supercomputer Exegesis - The DBC/1012, the NCR 3700, the Ynet and the BYNET," *Emerging Trends in Database and Knowledge-Based Systems*, IEEE Computer Society Press, pp. 139 - 157.
12. L. Gallagher, "SQL Generic ADT packages," *ANSI/X3 529-R*, American National Standards Institute, New York City, New York.
13. M. R. Genesereth and S. P. Ketchpel, "Software Agents," *Communications of the Association for Computing Machinery (CACM), Special Issue Intelligent Agents*, Vol. 37, No. 7, July 1994, pp. 48-53.
14. D. Mandelkern, "Graphical User Interfaces: Next Generation," *Communications of the Association for Computing Machinery (CACM), Special Issue Graphical User Interface (GUI)*, April 1993, pp. 36-39.
15. M. Stonebreaker, "The case for shared nothing," *IEEE Data Engineering Bulletin*, Vol. 9, No. 2, pp. 4-9.
16. A. Biliris and E. Panagos. "EOS User's Guide," *Release 2.0 Technical Report*, AT&T Bell Laboratories, May 1993.
17. M. Carey, D. DeWitt, and J. Naughton, "The 007 Benchmark," *Proceedings of the Association for Computing Machinery (ACM) Special Interest Group on Management of Data (SIGMOD) Conference*, Washington, DC, June 1993, Vol. 22, No. 2, pp. 12-21.

Warren Sterling *is director of multimedia projects at AT&T Global Information Solutions (AT&T-GIS) in El Segundo, California. He joined the Teradata Corporation (now part of AT&T-GIS) in 1982 and has been responsible for the development of the last three generations of the DBC/1012 hardware and a number of client products. His research interests are massively parallel processing systems, digital image processing, heterogeneous distributed databases, and multimedia databases. He received his B.S.E.E. degree from the University of Illinois, Urbana, and his M.S.E.E. and Ph.D.*

degrees in electrical engineering from Carnegie-Mellon University in Pittsburgh, Pennsylvania.

***Felipe Cariño** is co-founder of Teradata Corporations' (now AT&T-GIS's) Advanced Concepts Laboratory in El Segundo, California, and is a consulting staff scientist at AT&T-GIS. His research interests are in distributed and parallel processing, the UNIX* system, heterogeneous distributed databases, and multimedia databases. He received B.A. degrees in computer science and mathematics from New York University in New York City, an Executive M.B.A. from the University of Southern California in Los Angeles, and an M.S. degree in computer science from New York University.*

***Catherine Boss** is a senior consulting analyst in the Computers and Communications Center of Excellence at AT&T-GIS in Naperville, Illinois. She is responsible for the multimedia server architecture team developing software and communication solutions. Ms. Boss received her B.A. degree from the University of Michigan and her M.S. degree from the University of Wisconsin, both in computer science. She joined AT&T in 1980.*