Oracle Media Server: Providing Consumer Based Interactive Access to Multimedia Data

Andrew Laursen, Jeffrey Olkin, Mark Porter
Oracle Media Server Development
alaursen@oracle.com, jolkin@oracle.com, maporter@oracle.com

1. Abstract

Currently, most data accessed on large servers is structured data stored in traditional databases. Networks are LAN based and clients range from simple terminals to powerful workstations. The user is corporate and the application developer is an MIS professional.

With the introduction of broadband communications to the home and better than 100-to-1 compression techniques, a new form of network-based computing is emerging. Structured data is still important, but the bulk of data becomes unstructured: audio, video, news feeds, etc. The predominant user becomes the consumer. The predominant client device becomes the television set. The application developer becomes the storyboard developer, director, or the video production engineer.

The Oracle Media Server supports access to all types of conventional data stored in Oracle relational and text databases. In addition, we have developed a real-time stream server that supports storage and playback of real-time audio and video data.The Media Server also provides access to data stored in file systems or as binary large objects (images, executables, etc.).

The Oracle Media Server provides a platform for distributed client-server computing and access to data over asymmetric real-time networks. A service mechanism allows applications to be split such that client devices (set-top boxes, personal digital assistants, etc.) can focus on presentation, while backend services running in a distributed server complex, provide access to data via messaging or lightweight RPC (Remote Procedure Call).

2. Introduction

Simple, affordable access to multimedia information is both an enormous business opportunity and a powerful vehicle for people to change the way they live and work. Whether in the realm of shopping, news, movies, education or other applications, consumer multimedia will make obsolete much of what we know about storing, retrieving, and processing information.

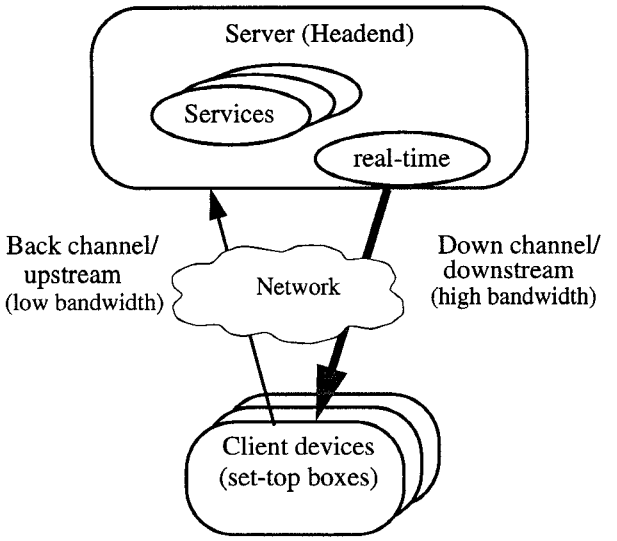
Providing information to consumers on a large scale presents many challenges. Ultimately, providers of information must find profitable means to reach price points that will drive demand, keep pace with technology, deliver simple access to computerphobic consumers, and deliver a robust architecture that allows the systems to evolve and grow.

The last decade of computing has produced inexpensive client hardware with shrink-wrapped software, scalable server hardware with complex data management software, and ubiquitous heterogeneous networking hardware with sophisticated networking software. However, the promise that data will be readily shared and easy to access has been mostly unfulfilled. Most software is single user and fairly easy to use or it allows resources to be shared but the degree of sharing extracts a correspondingly high price in usability. Consumer-based interactive networking is an attempt to provide simple access to unprecedented amounts of shared data. Oracle Media Server provides a framework for such an endeavor.

SIGMOD 94- 5/94 Minneapolis, Minnesota, USA
© 1994 ACM 0-89791-639-5/94/0005..$3.50

3. System Overview

This section describes the architectural components of a consumer-based interactive network typified by the following diagram:



The server consists of any number of computers networked in any fashion. The networks connecting clients and servers are asymmetric, with high bandwidth available in the downstream direction. Client devices today are generally built for interactive TV and are generically termed set-top boxes. Other classes of devices (personal digital assistants, video phones, etc.) will become important to consumer-based networks in the near future.

3.1 Networks

The major characteristic shared by the networks currently being deployed for interactive TV is asymmetric bandwidth. Downstream bandwidth — link from server to client device — ranges from a minimum of 1.5 megabits/sec (DS1 data rates) to 45 megabits/sec (DS3 data rates). Upstream bandwidth — link from client device to server — may be more modest, ranging from 9600 bits/sec to 64 kilobits/sec. Bandwidth will increase in both directions, but will probably remain highly asymmetric since most information flows toward the consumer.

Three types of physical transport define the three types of networks that are being deployed:

- *ADSL* (Asymmetric Digital Subscriber Loop) provides 1.5–6 megabits/sec of downstream bandwidth — up to 64 kilobits/sec in the opposite direction — over a twisted pair of copper wires. The maximum distance achieved to date is 6,000 meters. This technology provides broadband capability to millions of consumers over their existing phone lines.
- *Coaxial cable* promises to provide 500 channels [1]. Cable pro-

vides 450–1000 megahertz of bandwidth. State of the art radio frequency (RF) modulation technology provides up to 8 bits per hertz. A typical cable plant buildout for interactive TV will provide back channel bandwidth in the 5–50 megahertz range, analog broadcast from 50–350 megahertz, with the rest dedicated to single user downstream channels in the 3–12 megabit/sec range. Thus, each interactive user will be allocated a virtual downstream channel in the high frequency spectrum and a corresponding back channel in the low frequency spectrum. The downstream channel is typically a time-division multiplexed bit stream modulated over an analog channel (6Mhz for NTSC and 8 Mhz for PAL). There are, of course, many ways to allocate and modulate the available bandwidth and capacity can be greatly increased with the addition of fiber.

ATM (Asynchronous Transfer Mode) to the home over fiber/coax hybrid networks provides maximum flexibility for both bandwidth and addressing. Since ATM provides dynamic allocation of bandwidth, it is possible to provide variable bandwidth bit streams. Thus, it is possible to serve movies at 3 megabits/sec, sporting events encoded in real-time at 8 megabits/sec, and HDTV compressed at 20 megabits/sec (or whatever bandwidth is required). ATM also provides flexibility at the headend since many servers can be directly connected into the ATM switch fabric at very high bandwidth; 155 megabit/sec is available today, increasing to 622 megabits/sec or 1.2 gigabits/sec over time.

Another important network characteristic is latency. In today's systems, a message may take more than a second for a round trip. Back channels are slow, multiple hops are often required, and return values (bit maps, executables, etc.) are often large. Applications must be architected with these latencies in mind.

The transport protocol over the network depends on the direction. Upstream, the transport may be X.25, UDP, RS232, etc. Downsteam, the transport may be MPEG-2 (Motion Picture Experts Group) transport packets, MPEG-1 bit streams, UDP packets, or another high-speed protocol.

3.2 Compression

Video compression technology reduces large bandwidth video data to rates that can be supported by the interactive networks. Just as the new networks provide the essential hardware technology for interactive deployment, new methods of video and audio compression are the essential software technology.

Although, studio-quality digital tape drives deliver more than 200 megabits per second, a typical home wired for interactive service will have only 1.5–6 megabits per second of downstream bandwidth. Thus, content must be compressed at over 100:1 to be transported over the network. This high rate of compression cannot be achieved without encoding loss; some data must be thrown away. Compression-decompression algorithms (codecs) differ in how they choose what data to throw away and what to keep. Compression schemes, for instance JPEG (Joint Photographic Experts Group), apply frequency transformations such as DCT (discrete cosine transform) to the data which take advantage of spatial redundancy (adjacent pixels tend to be similar in frequency), Unfortunately, in order to produce a picture of comparable quality to a VHS VCR, JPEG bit rates of at least 4 megabits/sec must be used. This rate is too high for the low end of the network delivery bandwidth.

Fortunately, a video stream also has large amounts of temporal redundancy (adjacent frames are the same or similar). Compression algorithms such as MPEG-1 (Motion Pictures Experts Group) and MPEG-2 take this redundancy into account and compress it out of the data. Carefully compressed movie footage using MPEG-1 at 1.5 megabits/second is comparable in quality to a VHS VCR; at bit rates of around 4 megabits/sec, MPEG-2 is comparable to a laser disk; at data rates of 6–8 megabits per second (within reach of both ADSL and cable coaxial systems), the video quality is better than laser disk.

While video compression benefits from both temporal and spatial redundancy, audio compression does not. Thus the audio compression achieved for CD-quality sound is only around 7:1 (compared to greater than 100:1 for video).

Digitally compressed video has many advantages over traditional analog video. For example, it provides play-through compatibility between disparate standards such as PAL and NTSC, absolutely perfect frame stills, and perfect multi-generation reproduction of master content[2]. Digital compression offers features that are impossible with analog recordings, such as MPEG-2's scalability enhancements which allow users to pan across large pictures, and even zoom in on features, with an increase in detail.

Digital compression has disadvantages as well. By compressing the temporal redundancy from a stream, the resulting single frame of compressed video is no longer self-describing, relying on nearby frames to completely reconstruct its contents. This makes random access into the middle of a stream difficult; the jump can only be made to well-defined access points, of which there are typically two per second. This inter-frame dependency also makes it difficult to edit compressed video. Finally, because of the complexity involved in determining the dependencies between frames, it is very difficult to compress video in real-time. Though real-time video encoders exist today, they are expensive and not likely to break into the consumer market soon. In the meantime, video conferences from the home will be limited to the use of existing picture-phone technology. Note, however, that real-time decompression is simpler and can be accomplished by a handful of chips.

Fractal compression is a proprietary video compression technology that relies on the inherent duplication of basic shapes in pictures. Though currently still in the research stage, this technology promises very good compression ratios for certain types of data. Fractal compression does not eliminate inter-frame temporal redundancy, and thus does not have any of the drawbacks mentioned above. Computationally, however, it is very expensive.

There are other proprietary compression-decompression algorithms such as TrueMotion, and Indeo. Since large vendors are justifiably hesitant to use proprietary codecs, telephone and cable companies are supporting the MPEG algorithms to the exclusion of all other compression technologies. Eventually, the industry will settle on an envelope protocol that will transport any compressed digital data, regardless of format. This envelope format may be MPEG-2 transport, or a hypermedia format such as MHEG or

1. The concept of 500 channels is misleading; there will be broadcast channels and interactive channels. Typically, an interactive channel will be used by only one consumer.

2. Production studios are currently analyzing the effects of the consumer's ability to make perfect master copies of content. MPEG-2 has placed encryption and copy protection features inside the transport layer.

Hytime, or it may be something not yet invented.

Even though digital media compression enables real-time stream delivery over the bandwidth available in current networks, it is not a "throw-away" technology. As network bandwidth increases, it still makes sense to continue to compress the video and audio, and use the spare bandwidth to carry more information (simultaneous data with video, multi-way video conferencing, etc.).

3.3 Devices

A set-top box is a device that combines the functionality of current analog cable converter boxes (tuning and descrambling) and computers (navigation, interaction, and display). The current generation of set-top boxes have four major components: a network interface, an MPEG decoder, graphics overlay, and a presentation engine.

- The *network interface* provides both downstream and upstream interfaces over one or more physical connections.
- The *decoder* converts MPEG encoded data into audio and video. In addition, the MPEG subsystem may demultiplex application and control data from an MPEG transport stream.
- The *graphics overlay* provides at least one graphics plane, bitmap operations, and optional chromakey mixing[3].
- The *presentation engine* consists of a CPU, at least two megabytes of memory, and a lightweight, real-time operating system. The client portion of the application runs in this subsystem. The application is controlled through the use of a simple remote control with buttons or a joystick. There is no keyboard in the basic system

In order for interactive consumer systems to be widely deployed, vendors are targeting such set-top boxes in the $200-$400 range, and thus comparable in price to a low-end VCR. While vendors are currently building entry-level set-top boxes, higher end systems are envisioned, with better graphics capability, high speed printers, graphics and video capture, and perhaps keyboards for power users. Ultimately, the entire range of set-top boxes will be distributed as consumer devices, just as VCRs, video game players and TVs are today.

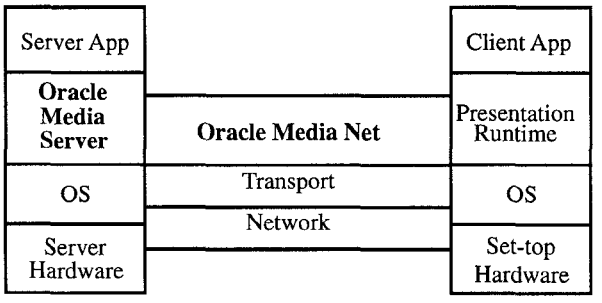
3.4 Data Types

Unlike most corporate data, the data being managed in these systems is mostly read-only. The data that is changeable is inherently partitionable (consumer preferences, PINs, etc.) or append-only data (billing records, usage data, etc.)., The amount of accessible data is several orders of magnitudes greater than that contained in corporate databases today.

The types of data that must be accessible to consumers include: isochronous, textual, structured, and binary large images.

- *Isochronous data*[4] is what comes to mind when the term multimedia is used. Films, television, and music will be online and available on demand. The main attribute of this type of data is that it is too large to simply download and store —it must be delivered in real-time with minimal buffering. Real-time delivery of audio and video is necessary for the success of consumer-based networks, but certainly not sufficient.
- *Online textual databases* will provide access to many terabytes of data, from live news feeds to current novels to popular magazines. One major problem is to navigate and search this sea of text using devices that have no keyboards. Oracle Media Server has the ability to reduce and abstract text which will help reduce the amount of text that must be initially displayed.
- *Binary large objects* (BLOBS) will be used to store many kinds of information, from images to application logic as stored in scripts. The same transport (with the addition of forward error correction) that is utilized for isochronous data is ideal for BLOBS.
- *Structured data* — as stored in a relational database — will be used much as it is today in corporate databases, to provide flexible access and to ensure integrity. Very large objects, isochronous and textual data will be inefficient if stored in conventional relational databases due to the way the data is searched and accessed. For example, an MPEG-encoded movie would not be stored in a relational database but all its attributes (director, leading actors, price, etc.) would be.

4. The Oracle Media Server Architecture



The Oracle Media Server provides a layer of software that enables distributed client-server computing in the consumer-based networks described above. The main components include:

- a service infrastructure
- comprehensive set of services
- access to data
- a real-time stream server
- messaging and RPC via Oracle Media Net.

The Oracle Media Server architecture assumes that data is stored and managed on the server and the client device provides a view onto that data. Generally, the view is through some paradigm such as a digital mall, personal digital assistant, or electronic newspaper. The user navigates locally and data is requested from the server as necessary. This provides a very clean split between the client side of the application and the server side.

Server applications (services) are "data based" and developed with the same tools used to build corporate databases — data modeling tools, schema editors, etc. These services must be built in a reliable and scalable manner.

Client applications are built using interactive, graphical authoring tools that allow digital assets (video, images, sounds, etc.) to be mixed with presentation logic to produce a run-time environment for interactive TV. Oracle Media Server will support runtime environments by providing such services as application download, asset management, authorization, and stream interaction (all described below). We envision that there will be many more client applications built than those for the server. For

3. Chromakey mixing allows video to be overlaid onto a particular color in the graphics plane (broadcast news weather maps use this technology).
4. Isochronous refers to data that has a time element to it (such as audio or video streams). The term real-time stream is synonymous.

instance, there may be three home shopping services nationwide that are used by hundreds of client applications.

Latency is a major issue in the network. Therefore, the architecture allows distributed applications to be built where all state is maintained on the client device. Lightweight RPC was chosen over traditional SQL for data access to both reduce the number of round- trip messages and to provide easy to use interfaces to application services.

4.1 Service Infrastructure

Each service comprises one or more cooperating server threads which may be distributed across several machines. Load balancing across these server threads is performed dynamically and transparently to the client. Requests sent by a client are routed to a service control point which decides, based on current system activity, which server process can actually handle the request. The service control points can support many server threads before they become bottlenecks. Still, because these systems must be highly scalable, service control points may be replicated. For example, the name service, which is used to locate all other servers, must be replicated in order to handle the large numbers of requests sent to it.

To shield client applications from network implementation details, they never interact directly with the underlying databases. Instead, all applications developed with the Media Server send messages to services — both the standard services that ship with the Media Server and customer-developed services Oracle7— using Media Net.

Interfaces to the services are defined using an interface definition language (IDL). An interface consists of a set of operations that define what the service can do. Once created, the interface definition is compiled to generate stubs which isolate the distributed nature of the system from the computations being performed. Client applications execute the operations by making remote procedure calls to the server.

In addition to providing its base functionality to client applications, each service has a standard interface for configuration, management, monitoring, debugging, logging, and auditing.

4.2 Access to Data

Whereas client applications access data via RPCs to services, services access data via a set of access libraries to the various data repositories:

- *Isochronous data* is stored in the Oracle Media Data Store, a realtime striped file system that allows concurrent, random access to video and audio. The interface allows streams to be positioned and played. All attributes describing the streams (title, content description, compression format, etc.) are stored as structured data in the Oracle7 database.
- *Textual data* is stored in the Oracle Text database as a set of indexed documents. The interface allows documents to be searched by words, phrases, and even concepts. The ability to abstract text is provided by Oracle*Context, a product that can parse and interpret English text using a sophisticated lexicon and 50,000 parse rules.
- *Binary Large Objects* are stored as opaque types in either the Oracle7 database or in the Oracle Media Data Store. As with Isochronous data, all attributes of the BLOBS are stored as structured data.
- *Structured data* is stored in the Oracle7 database and accessed via SQL and PL/SQL (Procedural Language/SQL). Oracle7 provides distribution, replication, and parallel access to the data. Stored procedures executing within the protected space of the database environment provide an excellent mechanism for building reliable services. A server application in this model consists of a schema and a set of procedures to access the data in the schema (which may be invoked directly from the client via the procedural service described below).

4.3 Core Services

The Media Server provides a set of core services from which other services can be developed:

- Isochronous service: Manages video and audio streams.
- Connection service: Creates and manages the connection to the client.
- Authorization service: Provides application security via PIN validation.
- Relational service: Provides access to the Oracle7 server and traditional relational data.
- Procedural service: Provides access to PL/SQL procedures from the set-top.
- Download service: Provides non real-time download to the set-top of new executables, data streams, etc.

4.4 Application Services

Application services provide the interfaces to the data used by the clients. They are developed by Oracle, network providers (telephone and cable companies), and third parties. Following are descriptions of example services (of which, there will ultimately be hundreds):

- The *video finder* utilizes structured data that describes all video content on the server. A movies-on-demand application running on the client would use this service to help the consumer select something to watch, utilizing such attributes as genre, director, actors, subject matter, etc.
- The *shopping service* utilizes a combination of structured and isochronous data to represent the electronic catalog. This service provides access to inventory, pricing, sales, etc. Transaction support (buying, credit authorization, etc.) is also supported by this service.
- The *news service* utilizes a combination of textual, structured, and isochronous data to represent news. Client applications using this service provide virtual news feeds based on consumer interest.

4.5 Administrative Services

Providing just video and interactive services to the consumer is not enough. Imagine if the telephone company provided all the wires but not repair, information, billing, upgrade or installation. Network providers must provide these services as an essential part of supplying interactive consumer access.

The Media Server provides a set of administrative services for managing users, billing, machines, and networks.

- System monitoring service: Provides overall monitoring of the system.
- Network monitoring interface: Allows communication with the network hardware, both for input to and output from the server.
- System management service: Provides the ability to bring pieces of the system up and down, change parameters, etc.
- Billing service: Provides a global way for other parts of the system to bill the customer for services; this may reside inside the system or merely call out to an existing billing service.

5. Real-time Stream Server

Providing isochronous data access is an inherently different

problem from traditional types of data access that lack the real-time component. Therefore, the real-time components of the Media Server are segmented from the other parts by a scheduling "fire-wall". All access into the real-time section of the server goes through scheduling dispatchers which analyze the load any given request will make on the system, determine if the request can be granted given the current system load, and then schedule the access. The real-time scheduler takes CPU, disk and memory resources into account when analyzing a request.

Even with the scheduler ensuring that the system does not become over-committed, there are many constraints on the real-time server design and operation. It must:

- service a large number of concurrent streams, each with independent control
- be reliable through any reasonable hardware or software failure
- store an enormous amount of data and coordinate movement of that data between different media and different servers
- allocate bandwidth between parts of the system
- be portable to any viable server hardware platform.

5.1 Stream control

The real-time stream server provides full VCR-like controls to the user: fast forward and rewind, slow forward and rewind, frame advance and rewind, random positioning, etc. However, it is not simple to fit these features into a real-time scheduling system, since each places a different load upon hardware resources. In addition, depending on the video/audio codec being used, each of these special modes may place different demands upon the real-time scheduler.

In the simplest model, a stream is merely a string of contiguous bits which must appear at the decoder at a particular rate. The server is responsible for accepting a command to start a stream and one to prematurely terminate a stream. In this model, streams would be easy to allocate, schedule, and deliver. Maximum service levels could be easily computed.

Unfortunately, the real world is not so simple. During the playback of a stream, many events may require the stream server to change its behavior. For example, if the user presses the pause button on the remote control, the server must stop delivering bits temporarily. Since there are network buffers between the server and the set-top box, the server has no way of determining exactly where the set-top box was paused. So the server and set-top box must communicate so that the server may queue from that point so that it will be ready for the subsequent play command.

The other VCR control commands are even more complex to implement. They all involve changes of bit rate, hard-to-predict disk seeks, and large amounts of CPU processing. Each of these must be factored into the real-time scheduler in order to provide the maximum level of continuous guaranteed service.

As described above, there are multiple video compression methods. For each compression-decompression algorithm, a software module handles all the aspects of that algorithm. This module provides entry points not only for all the different rate controls, but also provides entry points to allow the caller to reduce or enlarge the spatial resolution, temporal resolution (frame rate), tunneled data rate, etc. This modular component allows the real-time server engine to stay independent of which codec is used for each stream.

By providing random entry ability and controlled playback of data, the stream service not only provides rich real-time stream playback but also is part of the hypermedia interactive environment provided by Oracle Media Objects[5]. Thus, in addition to playing movies at a user's request, the server can play video or audio clips in response to other services, such as home shopping, education and video hyperlinks.

Stream control features make the consumer experience superior to that provided by normal videotape in many ways. Users can choose movies in minutes without leaving their houses. They can change their minds half-way through a movie and select another. They can stop a movie half way through and choose to continue it an hour, day, or year later, the digital content never becomes worn or fuzzy from overuse, and the digital frame still and advance capabilities are equivalent to those found only in very high quality VCRs. Finally, studies have shown that what consumers like least about renting movies is having to return them. In fact, people dislike this so much that late return fees are a significant, budgeted percentage of a video rental store's income.

5.2 Reliability

As information servers take on more and more of the communication tasks of consumers, they will have to become as reliable as the current telephone system. To achieve this, the real-time server can be configured for various levels of reliability.

By far the most common failure point in the isochronous server is a magnetic disk drive. This is largely because the disk drives are the only part of the core system with moving parts. Even with mean time between failures (MTBF) of 1,000,000 hours, systems with large farms of disks will experience individual disk failure on a regular basis. For a system with many terabytes online, and thousands of disk drives, frequent disk drive failures could render large amounts of content inaccessible.

The real-time server can identify and correct disk failures without interrupting the real-time data flow. The approach used is unique (and unfortunately proprietary, so the details cannot be discussed in this paper) since traditional approaches to disk redundancy (e.g. RAID) are anything but real-time. Disks also go offline many times during the day due to thermal recalibration, predictive failure analysis checks, etc. As with disk failures, the system will ride through these temporary interruptions without any interruption of real-time service. On any system with disks that may be hot-swapped, when the Media Server has determined that a disk has failed and needs to be replaced, it prompts the operator to remove the bad canister and load in a new one. It then requests resource bandwidth from the scheduler to rebuild the volume as fast as possible without disturbing stream playback. The system provides two dimensions of reliability configuration for disks; the system manager can select what storage overhead to incur to guard against both multiple simultaneous disk failures or multiple disk chain controller failures. The ability to continue uninterrupted play through both disk and controller failures solves the largest reliability challenge present in the real-time server.

The second most common failure is a network backbone communications failure, most likely from a piece of hardware outside the real-time stream server. To cope with this, the system can accept routing commands during stream playback. When the network management service determines that a stream needs to be re-routed, it tells the network switch to re-route the signal and simultaneously informs the stream server where it should send the data. This happens on the fly, without having to restart the stream.

5. Oracle Media Objects is an authoring and runtime environment for client devices whose discussion is beyond the scope of this paper.

Software failure is a third potential source of system failure. For the most part, pieces of the system behind the real-time "firewall" are independent from each other, so a single software failure should interrupt at most a handful of streams. Because the real-time stream service checkpoints frequently, the stream can be restarted from a different point in the system quickly with only a small service disruption to the user. Once the service manager detects the software fault, it immediately restarts the parts of the system that failed.

Hardware and software redundancy and independent restartability are part of the basic architecture of the Media Server. By placing these elements into every service, a complex system can be diagnosed and restored to normal operation in the shortest possible time.

5.3 Storage Capacity

Financially, the key metric in serving video is megabytes per second per dollar (more generally, bandwidth per time per cost). Since different content (feature-length movies, classic movies, home shopping videos) will have different revenue and usage patterns, the server allows the data to be stored on a variety of devices, each with different cost, bandwidth, and capacity characteristics.

The system manages a multi-tiered storage system where movies might be staged from off-line tape to an online optical jukebox to magnetic disk and finally to RAM. With commodity disk drive prices currently at approximately $600/gigabyte, a 2-hour movie costs $800 to store on magnetic storage. This same movie costs $40,000 to store in RAM and about $50 to store on off-line tape. Where the content resides at any given time is determined by the server but a human server manager can over-ride that decision at any time. When a decision is made to move a piece of content, this request is scheduled into real-time bandwidth just like user requests.

In addition to staging data between multiple tiers of storage on a single server, it can also move between servers over a high-speed inter-server network backplane. Because of the design of the storage system, staging from another server is equivalent to retrieving from offline tape.When data is moved between different storage media or between servers, the reliable messaging features of Media Net are used.

When content is loading from a tape, the user is given control as soon as possible without waiting for the entire movie to load. While the movie is playing, content is cached in either disk or RAM to provide full rewind and still capabilities. Of course, since the movie is loading from a sequential medium, fast forward is limited to the amount the server has loaded ahead of the user's current frame.

5.4 Bandwidth Scheduling

Interconnect bandwidth is the limiting factor not only in the networks being deployed but also in the real-time stream server. Even with the high data and video bandwidths being used today, CPU power is plentiful. This bandwidth is divided into three categories: reading from external I/O devices such as disk or tape, internal routing of the data within the server, and the external routing of the data over the network. By balancing these three, the cost per stream can be minimized.

Building a well balanced system requires that it must be scheduled precisely, or one segment of it may temporarily become overloaded. As described above, the real-time scheduler has a very complex job. It must mix in bandwidth requests from streams in normal play-forward mode, others which are being fast-forwarded, disk rebuild processes, downstream Media Net application data, offline media requests, and requests to and from other servers. The real-time scheduler allows the system to be used within a small margin of maximum capacity, yet still ensures that no stream will ever glitch.

It is transparent to the administrator whether the system is serving thousands of streams of the same movie or thousands of streams of different movies; any combination is possible, and the manager does not have to make decisions about disk layout in order to achieve better performance for one load pattern or another. In addition, if the network connected to the server supports routing one stream to multiple households, the system can "piggyback" users onto other users already playing that stream at nearby points. Of course, this optimization becomes more complicated when one of the users hits the pause or fast-forward keys, because the server must start a new stream to service the customer who split from the concurrent pack.

5.5 Portability

Current server platforms have a 2–3 year product cycle. Because of this, the server code must be written portably, yet provide a high level of integration for each implementation.

By requiring only a low-level real-time kernel, the Media Server is portable among many existing and new hardware platforms. The Media Server can be delivered on a wide range of systems, ranging from workstations serving a handful of streams through symmetric multiprocessor (SMP) machines up to massively parallel (MPP) machines delivering many thousands of streams. In addition, since the system depends only on features that must be minimally present in all hardware, it is operating-system independent.

6. Media Net

Media Net provides the communication backbone that allows services scattered across heterogeneous, asymmetric networks to communicate with each other transparently. For the most part, the level of exposure of the network for application servers and clients is minimal, since RPC serves to hide these details.

Oracle Media Net is an implementation of layers 3 (network) and 4 (transport) of the OSI reference model. However, it does not seek to replace the existing functionality of other networks; instead, it augments their ability to deal with the network topologies and quantities of traffic common to consumer-based networks.

6.1 Network Layer

A Media Net network is composed of one or more underlying networks, each with their own characteristics. The topology of the network is described as a directed graph whose vertices are nodes (either intermediate nodes or endpoints) and whose edges are data links (OSI Layer 2).

A node is a point in the network through which data is routed. The mapping between a node and an operating system process or machine is arbitrary. A single process may allocate several nodes, and many processes on a machine may allocate nodes independently.

The edges of the graph are directed, which means that each data link specifies the direction in which data can be sent. Some data links are bidirectional. To a user of Media Net, however, the entire network appears bidirectional. That is, given any two vertices, if there is a cycle in the graph that includes the two vertices, those two nodes may communicate. Routing mechanisms hide the

unusual topologies.

Data links may either be packet-switched or circuit-switched. Using a packet-switched link, data may be directed to any of a number of endpoints by specifying its address. Using a circuit-switched link, data sent from one end of the data link can only arrive at the other end of the link. Circuit-switched links can be unidirectional or bidirectional. For example, X.25, ATM and TCP/IP are bidirectional circuit-switched links. A T1 line carrying data into a client device is a unidirectional link. UDP/IP and most interprocess communication (IPC) facilities are examples of packet-switched links.[6] In any case, a packet is used as the lowest common denominator of transport since a byte stream may easily be divided into packets.

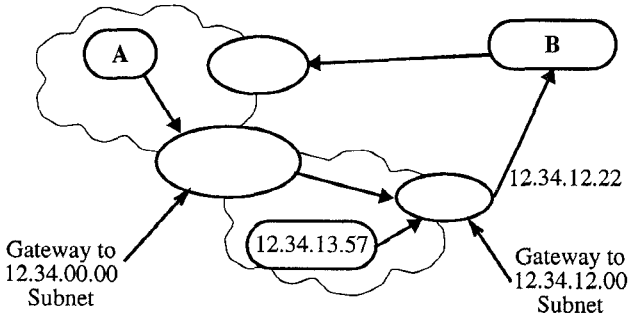
6.2 Routing

Because a packet may travel through several different types of underlying networks, each with their own addressing schemes, Media Net defines its own independent address space. This hides the many different types of addresses in use for each type of data link.

Each media net address is a 64-bit word that globally identifies an endpoint of communication, or node. The address maps to a directed edge in the network. When a node sends a message, it specifies as a reply an address which maps to the edge directed back toward the sender. This may be the same link over which the data is sent (if the data link is bidirectional) but is usually not.

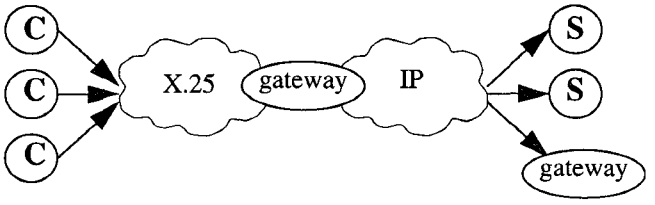
The routing problem is to find all of the intermediate links between a sender and the end point of the link at the given address. Since this is a recursive problem, the address space is divided up into a hierarchy of subnets. Routers need only know how to reach subnets. The knowledge of how to reach a particular node is distributed throughout the system, rather than centrally located in one place.

A prefix of the address is used to identify a subnet. The longer the prefix, the finer the granularity of the subnet. Subnets may be arbitrarily mapped to processes and machines. For example, a single process, may own an entire subnet.



In the figure above, node A needs to send a packet to the node at address 12.34.12.22. It first sends the packet to a gateway for subnet 12.34.0.0, which forwards the packet to a gateway for subnet 12.34.12.0. That gateway routes the packet to the destination endpoint. Note that an address really identifies a particular link, not a node, so one node may have several addresses. The clouds in the figure represent different types of underlying networks.

Routing decisions are made only when a packet moves across a junction from one type of data link to another. For example, since an IP-based network appears as one hop (see figure below), Media Net gateways need only be present when a packet must enter or leave the IP-based network. Since existing network routing need not be replaced, network resources can be used very efficiently. In a completely homogenous bidirectional network, no gateways would be required, and the routing functionality of Media Net would not be used.



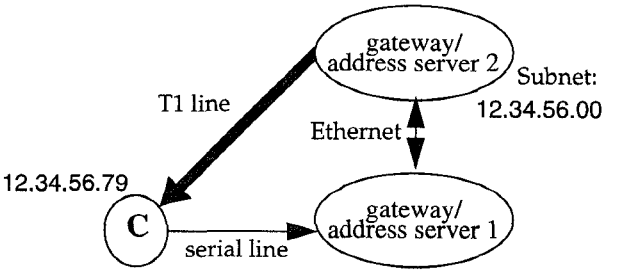
6.3 Address Requisition

Routing would quickly become an administrative nightmare if addresses had to be managed manually, especially in situations where large numbers of client devices enter and leave the network with a high rate of turnover. Thus, all routing tables are managed dynamically using dynamic address allocation.

When a node is started, it must obtain a Media Net address to identify itself. Alternatively, it may have a well-known address that it needs to announce.[7] Requesting an address is complicated since the data links in the network may be unidirectional, and physical network broadcast (e.g. Ethernet broadcast) is not available in most cases (nor would it be feasible considering the number of nodes in the network).

In order for a node to obtain a Media Net address, it must know the data link through which it can reach an address server, and the link through which it expects the response. The latter downstream link is the one that is actually assigned the address.

The address request specifies the downstream link in a data-link specific manner. The problem is to figure out how to get to that data link. The information provided may not be sufficient to be able to complete the request. Often the address server that receives the request is not connected directly to the downstream link.



In the figure, suppose that address server 1 receives an address request for a client device over a serial line. The client specified a T1 line as the downstream link. Unfortunately, the T1 line is not attached to the machine where address server 1 is run-

6. It may seem unusual to refer to any of these protocols as a data link, since they all involve several layers of the OSI stack, but recall that Media Net does not replace existing network functionality. Thus, an IP-based network appears as only one hop to Media Net.

7. In practice, well-known addresses are used only for name servers and system processes that must continue to operate even after a catastrophic failure of the name servers.

ning; it happens to be attached to a different machine which is connected to the first via Ethernet. Address server 1 forwards the address request over Ethernet to address server 2 which allocates an address and issues the response through the downstream channel.

Media Net is able to forward address requests appropriately because gateways know about each other and about the types of data links to which they are connected.

Why is all this necessary? Because the address that is ultimately assigned to a data link must belong to a subnet that identifies the appropriate destination.

Suppose the client requested a second address, but this time specified the serial line (now bidirectional) as the downstream link. This time, address server 1 can assign an address itself since it is connected directly to the serial line. The client now has a second address completely unrelated to the first. This leads to an important point: only an address server attached to the head of a link can legitimately assign an address for that link.

As a very useful side effect, dynamic address allocation causes the creation and maintenance of routing tables. Because the address request is self-identifying (even if only in data-link-specific terms), the address server has enough information to enter new routes into the appropriate routing tables. Address servers themselves may requisition entire subnets from larger-scale address servers, thus serving to automate large amounts of network administration.

6.4 Reliable Messaging

Network links are assumed to have the usual array of problems: they can drop packets, deliver packets out of order, duplicate packets or corrupt packets. The transport layer provides reliable communication over such networks. Like the network layer, the transport layer does not replace or duplicate existing functionality. Thus, if the network link is actually reliable, Media Net can provide reliability cheaply, without using its own mechanism on top of an already reliable link. Of course, if the data travels over multiple links, and at least one of those links is unreliable, then the transport layer will revert to using its own reliability mechanisms for that data.

Reliability is implemented using a straightforward *positive acknowledgment with retransmission on timeout* strategy. However, the characteristics and usage of the network make it difficult to estimate timeout. Consider a T1 line with 1.5 megabits/sec capacity. All of the bandwidth may be available for data traffic until a user starts receiving video. Then, the bandwidth left for data is suddenly reduced to less than 20 kilobits/sec with typical MPEG compression.[8] At the other extreme, consider a PDA user who moves from wireless to wire-base communication by plugging in a phone jack. In this case, the bandwidth available jumps dramatically.

The transport layer must be able to adapt to these changes in order to maintain reliable and efficient transport. TCP/IP's means of solving these problems are ineffective because they rely on the existence of a connection. Since most data traffic in these networks is *connectionless* (for reasons explained below), another solution must be used.

The principle behind the solution is the same as for TCP/IP: If the round-trip time—the time elapsed from when the message was sent to when an acknowledgment was received—diverges widely from the expected round-trip time, we can assume that the available bandwidth of at least one link has changed.

However, without connections, tracking round-trip times provides almost useless information. There is no reason to assume that the same round-trip will occur repeatedly. Nor does it help any of the other nodes in the network. Instead, Media Net focuses on trip times across individual links. This information is propagated through the network over time to allow individual nodes to have local access to trip estimate information. To avoid saturating the network with control messages, this information is usually piggybacked inside data messages. As a benevolent side effect of the propagation, brief, spurious fluctuations in bandwidth have no noticeable impact on trip estimation.

7. Conclusion

Whether the much-hyped information highway described in the popular press materializes this decade or next is unimportant. What matters is that a network infrastructure is being deployed today that enables consumers to gain considerable interactive power from their homes. This infrastructure is aided by new video compression algorithms which provide high quality video and audio at low bit rates yet can still be decoded very inexpensively in the home. Together they form a business case to provide hardware in the home, along with software, distributed across the network in a variation of the traditional client-server model. The business case depends on providing a mix of current media products (movies, music, etc.) in order to generate revenue with more visionary applications. These applications will be designed and refined until the system forms a complete model of consumer interactivity and information gathering and filtering.

The Oracle Media Server provides an architecture in which this consumer model can evolve. By offering a service framework that supports server application development, the Media Server allows developers to concentrate on the design of the media-based applications, such as interactive shopping, news, games, research and education. Oracle Media Net enables these applications to communicate transparently across the complex asymmetrical networks of today. Behind the services architecture, an isochronous server and traditional data access methods deliver an information-rich environment to the consumer via RPC.

Version 1 of both the Oracle Media Server and Oracle Media Net exist today and are being deployed in interactive TV networks over both cable and ADSL. Everything described above is part of this release of the software. Version 2 of Oracle Media Net will provide light-weight authentication on a per message basis and support for corporate networks. Version 2 of Oracle Media Server will provide an object brokering capability from the server.

8. Acknowledgments

The authors would like to acknowledge the remainder of the development team: Farzad Nazem, William Bailey, Jeffrey Sussna, Mark Moore, and Rob Langhorne.

Special thanks to John Zussman, William Bailey, Frank Chen, Mike Carey and Shel Finkelstein for their timely review of the paper and their excellent suggestions for improving it.

8. Because of the special real-time characteristics of media-based data, the transport of such data occurs effectively "outside" of a Media Net data link. However, its impact on data transport is very real.