Digital Video Recording And Video-On-Demand Servers

Class #330

Mike Ficco

Embedded Systems Conference
San Francisco, 2004

Introduction

Video-On-Demand (VOD) has long been an unfulfilled dream. However, the convergence of new technologies and economical extremely large hard drives may finally produce a compelling business model. The challenge now is for settop box and other embedded system developers to deliver on this long promised service. This paper discusses various aspects of building a VOD Server based on Digital Video Recording technology. It assumes some familiarity with embedded systems.

Video-On-Demand History

Early attempts at VOD mostly failed because the great attraction of the system became its downfall. That is, the promise of a large selection of videos available upon demand led to significant and economically unsustainable bandwidth consumption. A partial solution was Quantatized VOD (QVOD), where programs were assigned discrete start times a few minutes apart. This provided near VOD and helped greatly with the bandwidth problem by aggregating multiple viewers into a single broadcast. The QVOD scenario allowed the bandwidth cost to be amortized across a large number of customers. Unfortunately, this benefit was only realized for a relatively small number of movies interesting to a large number of people. The VOD service still could not afford to subsidize the bandwidth cost of large numbers of customers viewing obscure movies. Economic realities forced VOD services to restrict the available movies to primarily new releases and mass-market programming. The promise of any movie, any time remained elusive.

Enter Digital Video Recording

Digital Video Recording (DVR) brings a new paradigm to the world of VOD. With modern extremely large hard drives and sophisticated digital compression, the possibility of locally storing large numbers of movies becomes a reality. No longer does the user have to wait while he is “aggregated” into an economically large bundle of customers. Obscure and special interest movies need no longer consume prohibitive amounts of bandwidth. DVR enables the acquisition of an instantly available library of hundreds or perhaps thousands of movies. In the next few years, one could envision building and shipping a relatively low cost settop box preloaded with hundreds of movies. A cable or satellite connection to such a product enables revolutionary service opportunities such as the “100 Movies A Month Club”.

Reference Feature Set

In any project, the requirements should be the first thing established. In our case this corresponds to the feature set to be supported by our settop VOD server design:

- The settop should be able to record multiple different programs simultaneously. This may involve capturing a digitally encoded transmission, or in digitizing and compressing analog programs.
- The settop should have the ability to play (output) analog programming to a TV via NTSC/PAL, component video, s-video, etc. This output may include picture-in-picture of a different program.
- The settop should have the ability to play (serve) one or more streams of digital programming (multimedia) to a network connection.
- The settop should have the ability play multiple different programs simultaneously. Such play may include some combination of pre-recorded and “live” (i.e. just being recorded) programs.

- The settop should be able to perform all recording and playing functions at the same time. It would be acceptable that the sum of the number of record and play activities be limited to a maximum number. That is, one may choose more recordings at the expense of fewer plays.
- As suited to the current state of the program, the settop should be able to perform the normal functions we've come to expect from VCRs (pause, play, fast forward, rewind, etc.).
- The settop should have the ability to perform advanced functions only possible with hard disk based digital recording. These may include, for example, the ability to instantly reposition to another part of a playing program, to toggle viewing between two programs without missing anything, dynamically insert personalized information into a playing program, or perhaps to video edit recorded programs.
- The settop should have the ability to display a list of recorded programs and should allow the user to navigate the list and select a program to watch. List display and navigation may be local via the TV screen or remote via a network connection.
- The settop should provide the ability to delete, hide, and otherwise manage the recorded programs.

Embedded Platform Architecture

Overview

This paper focuses on the embedded components and modules directly related to DVR and VOD functionality in a digital TV/movie network. Hence, it is assumed that such critical modules as the power supply, tuners, network interface, front panels, infrared receiver, etc. simply exist and work. Obviously a great deal of variation is possible in the design and implementation of these modules. We assume that such variation is constrained to remain compatible with the goal of DVR and VOD.

Major Modules

One thing that becomes apparent from the feature list is the need for segregating the record and playback functions. It is critical that recording of a program continue without interruption while the user hops from viewing (playing) one program to another. As shown below, there are only a few modules specific to DVR/VOD. In effect, the Record Module, Hard Drive, and Play Module are wedged between the normal connection of the Transport to the CODEC. Of course there are likely to be a few differences in the implementation details of the Transport module and the CODEC module when modified in this fashion.

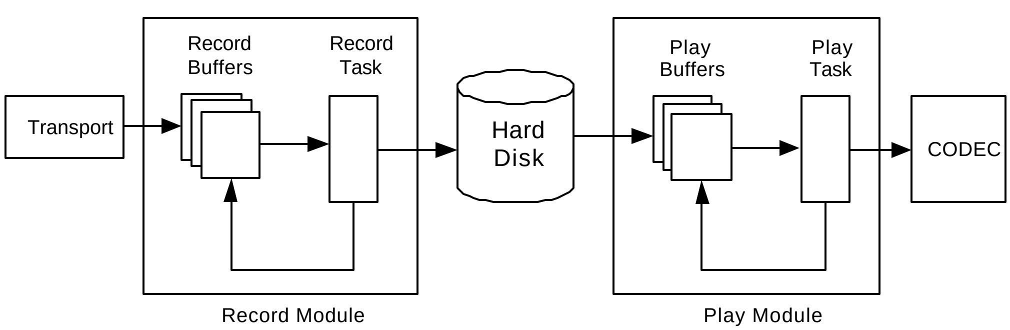

Figure 1. DVR/VOD Major Modules

Transport

This is the source of data for the system. In this paper we define the transport module to extract the digital bits relevant to viewing the program selected by the user. These bits may include audio, video, condition access, closed captioning, and other program related information. The data is passed to the record module where it is written in a suitable format to the hard drive.

Record Module

The job of the Record Module is to “massage” the transport data as needed and store it in a file. Additional relevant information (title, description, time of recording, etc.) may also be stored along with the audio and video bits. Our design should accommodate multiple transports, multiple hard drives, and numerous simultaneous recordings in a high-end system. In such a scenario, it is reasonable that a single Record Task exists for each hard drive since multiple Record Tasks would still have to synchronize their access to each disk. Furthermore, to simultaneously record ‘n’ programs we need ‘n’ open record files and ‘n’ distinct transport streams. A single pool of record buffers shared among all tuners and all hard disks is possible since a record buffer can never "hang". Even in the case of a disk write failure, the buffer will eventually be returned to the record buffer available queue. In addition, variations in the buffer usage rate (due to differences in the bit-rate of each transport stream) actually benefit from the statistical leveling that occurs from allocating all record buffers from the same queue.

Hard Disk

In the interests of efficiency, both the write (record) and read (play) interfaces to the hard disk should be via Direct Memory Access (DMA). DMA relieves the processor from this labor intensive activity and speeds transfer of the data. A well designed VOD embedded system will insure that the hard disk is ultimately the digital media bottleneck. However, a good file system, efficient hard disk device driver, and correct buffering should enable a single hard drive to service multiple simultaneous record and playback multimedia streams. By far the greatest delay in accessing a hard drive is moving the drive head to the correct position. Efficient use of the hard disk requires that data be written and read in large contiguous blocks to minimize head motion. However, large disks blocks may increase channel change time. The optimum size of the block depends on many factors and should be carefully analyzed early in the development cycle.

Play Module

The job of the Play Module is to periodically poll the file being played to see if there is data available. As a program is being recorded fresh blocks of data periodically become available. The Play Module reads that data into buffers and forwards it to the CODEC. In the case of a previously recorded file, data will be continuously available and the limiting factor will be the availability of buffers in which to read it. Since the original data may have been somewhat “massaged”, the Play Module must also “un-massage” the recorded data to the point where the CODEC can understand it. It would be simplest to have a distinct Play Task devoted to every file being played since coordinating potentially randomly available data from multiple files adds unneeded complexity to a single task. Tasks are cheap. Each file should have it's own Play Task to poll for data and send it out in distinct streams to CODEC or network destinations. Unlike the Record Module, a single pool of buffers for all files being played is a bad idea. There are two major problems with such an approach. First, play buffers <u>can</u> hang. This happens whenever the user pauses play. Second, in pause, slow motion, and other situations, it is reasonable to pre-fill available buffers and queue them for later processing by the CODEC. Indeed, this is the normal steady-state situation while playing back a previously recorded file. The easiest way to handle this is to have a separate pool of buffers for each file being played. When all play buffers in the pool associated with a file are filled, the Play Task sleeps until one is consumed and freed by the CODEC. With separate pools for each file we lose the statistical leveling advantage discussed previously. However, we gain significant simplicity in handling DVR “trick modes” like fast forward and rewind.

CODEC

The CODEC consumes the multimedia data and presents audio and video to the user. In our view of the world, the CODEC can be resident on the settop and output NTSC/PAL data to a connected TV. The CODEC can, by the miracle of modern telecommunications, be at some remote location half the world away. This paper is not particularly concerned with details of the CODEC. The assumption is that the Transport and CODEC previously worked and we inserted the Record/Disk/Play modules into a working system. Having said this, it is likely that some modifications will have to be made to the CODEC. One reason is that the character of the data may change somewhat due to periodic bursts of blocks of data read from the disk. You may also want to add features to the CODEC to take advantage of the new capabilities enabled by the addition of digital recording.

System Issues

So far we may have created the impression that you take a working digital settop, pop in a hard disk, add a couple of tasks, and BINGO! You've got a DVR/VOD system. Unfortunately it is a little harder than that. This section addresses some of the typical issues involved in making a DVR/VOD system work.

Task Priorities and Resource Allocation

A good system design allocates sufficient buffering, processor horsepower, disk bandwidth, etc. so no task ever starves of data or processing. The challenge of an embedded system designer is to minimize the system cost (memory size, processor horsepower, etc.) while maintaining adequate data processing capabilities. There are no special DVR/VOD challenges associated with establishing the task priorities. General practice for embedded data acquisition systems is acquisition supersedes distribution. In effect, DVR is a multimedia data acquisition system. Hence, record tasks have higher priority than play tasks. With respect to the system, recording should be one of the highest priorities but the Transport should be still higher. Trying to record when you are receiving no data is pointless. An issue that comes up frequently in VOD applications is guaranteed availability of sufficient system resources to assure that a new client can be fully serviced. This is not an appropriate consideration in a consumer product. <u>Guaranteeing</u> that sufficient worst case resources are available is extremely pessimistic. The world will not end if once every two days the stars align to cause a momentary resource shortage. Denying new clients that could be well served under typical circumstances is unacceptably pessimistic and costly for a consumer product. A more appropriate consumer product model is that of graceful degradation. When system resources get tight, play pauses but recording continues. Setting the criteria of acceptability is a difficult responsibility. Certainly your customers would be unhappy if their movie stopped every few minutes. Some careful system analysis and consideration of the production cost of the product and your corporate image is involved in setting this quality level.

State Manager

There are a surprisingly large number of operational states in a DVR/VOD system. In general, the State Manager is involved whenever the multimedia stream to a CODEC is changed or disrupted. Examples of this include selecting a new program to watch and fast forwarding through a commercial. Most such states are associated with PLAYING multimedia. Managing recording is fairly simple… start recording - stop when done. Hence, we observe that the State Manager should have a priority higher than any play task but lower than any record task. We further conclude that it would be best to have a State Manager for each Play Task. This is because changing a channel, flushing queues, or reinitializing a CODEC can involve substantial delays. It is simplest (and therefore more likely to work correctly) if a separate task is devoted to managing each one of these scenarios.

Audio/Video Synchronization

Audio and Video flow through different hardware subsystems on their way to being presented to the customer. Hence, audio and video are generally split apart and separately managed by the CODEC. In a well designed digital TV network, associated audio and video arrive from the Transport at approximately the same time. This means that connecting the Transport directly to the CODEC will lead to approximate audio/video synchronization. "Approximate" is usually not what is desired and companies tend to exert significant effort to get the best possible A/V sync. Playing digital buffers recorded to a hard drive adds some additional challenges to this since the "approximately" associated audio and video coming out of the Transport may be recorded to the disk in physically different blocks. The separation could be several blocks depending on the type of compression, digital network design parameters, the size of the disk blocks, and multiple additional issues. The DVR/VOD embedded system design must accommodate the worst case system and network A/V skew. It may also be desirable to consider muting a program and how to handle music channels (i.e. audio only) in your design. Finally, features like fast forward do not usually present audio while showing the high speed video. This must be accommodated in your buffer handling and flushing when switching play modes.

Channel Change

Channel change on a DVR/VOD enabled settop is remarkably complex and deserves some special consideration. The first thing to note is that there is a conflict between efficient disk utilization and rapid channel changes. This is because efficient disk utilization depends on the writing and reading of large contiguous blocks of data. However, after a channel change, waiting for a "large block" of data to be accumulated may take a substantial period of time. Sending a copy of the data being recorded directly to the CODEC can diminish this problem by bypassing the hard disk subsystem. However, doing so results in more complex stream transitions in pause, fast forward, and rewind scenarios. Usually simple is best since complicated things are more likely to break. The mantra of embedded system design is "as simple as possible, but no simpler". The "no simpler" part of this phrase unfortunately means that you must occasionally resort to more complex solutions to meet the requirements. Carefully consider the product needs for the interrelated aspects of channel change times, A/V bit-rate, and the disk drive utilization.

Channel Change Algorithm Example:

- Stop recording A/V data coming out of the Transport.
- Start tuning to the new channel.
- Stop feeding "play" buffers to the CODEC.
- Tell the CODEC Module to black the screen.
- Tell the CODEC Module to stop decoding data.
- Tell the CODEC Module to flush its buffers.
- Wait (sleep) until the CODEC flush is complete.
- Tell the Record Module to flush its buffers to disk.
- Wait (sleep) until the RECORD flush is complete.
- Close files involved in recording or playing the old channel.
- Wait (sleep) until the Transport is locked onto the new channel.
- Open a File and associate it with the new channel.
- Enable playing this File (at first there will be no data).
- Enable recording to this File.
- Tell the CODEC Module to start decoding data.
- Tell the CODEC Module to un-black the screen.

Multimedia File System

Perhaps the most critical component of an embedded DVR/VOD system is a highly efficient multimedia file system. Unfortunately most file systems, especially those available for embedded applications, are not intended to handle multiple streams of gigabytes of data moving around at several megabits per second. As daunting as the prospect is, this may indicate the need for developing a custom file system for some years to come. In an embedded multimedia application, LOTS of data and data structures are changing all the time. Hard drive head motion is so slow that it is impractical that all these changes interact directly with the disk. The only efficient solution is to spend the RAM and keep the relevant information instantly available in memory. This information should be periodically backed up to disk to insure that recorded programs can be replayed after powering down the settop.

Initialization

Once the embedded system incurs the expense of a hard drive it is reasonable that other non-volatile storage in the system be eliminated or minimized to save cost. In this environment, the hard disk must have a structure that permits sharing these various storage needs. It is imperative that the system be able to correctly identify that a drive has been "partitioned" in this fashion. In particular, the multimedia “partition” must have a verifiable structure that contains a “directory” of recorded programs. In traditional file system nomenclature, each directory entry would be associated with a File Access Table (FAT) that allows all the contiguous blocks of disk storage to be sequentially retrieved and reconnected into the original stream of media. Continuing to use the traditional nomenclature, we “format” this multimedia partition by collecting all its unused blocks into a list or table and labeling it as Free Space available for use. Some form of checksum should be associated with an initialized and formatted disk to verify its status as ready to use.

File System Features and Attributes

The multimedia partition is formatted to allow storage of multiple types of information. The vast majority of the partition is allocated to storing the actual audio and video data. A portion of the partition is used to store the DIRECTORY and the FAT. Additional room may optionally be devoted to program related information such as a plot summary, date recorded, etc. At a minimum, the multimedia file system needs to understand the difference between available disk space (Free Space) and previously recorded space. As described in the following sections, much more sophisticated features can be added as needed or wanted.

Free Space

Upon formatting the multimedia partition, all audio/video storage space in the partition is declared to be FREE. As programs are recorded to files, the Free Space is consumed. When a program is deleted its storage is returned to the pool of Free Space. If your system expects frequent recordings and deletions Free Space fragmentation is certain to be an issue. Care should be taken to create well designed and highly optimized allocation, de-allocation, and de-fragmentation mechanisms.

Opening A File

A modern large hard drive can hold hundreds of multimedia programs. It is reasonable and prudent that system resources are only allocated to those files that are actually being used. A fundamental design criterion of your VOD server is to set the maximum number of files that can be simultaneously accessed. This leads to the reasonable and traditional requirement that a multimedia file must be opened before using it. At the time the file is opened, system RAM is allocated to the existing FAT (in the case of a previously recorded program) or to Free Space that will become the FAT of a program being recorded.

For proper operation, the VOD file system must associate with each open file several read pointers and an "open" instance counter to handle multiple simultaneous viewers of the same file. Of course, a file can be opened for reading (playing), writing (recording), or both.

Recording

When a file is opened for recording it is assigned the recording file type. Practical experience has shown it is useful to pre-allocate a large block of Free Space storage to a file when it is initially opened for recording. This allows recording to happily proceed without the burden of the allocation issues. Over time, recording gradually erodes the Free Space allocated to the file. A low priority background task periodically allocates additional Free Space when more is needed. Design issues for your system include whether Free Space pre-allocation is needed, the size of such an allocation, and how to handle the error condition of a full disk.

Program

When recording of a program is completed, pre-allocated but unused Free Space is returned to the main pool. The file type is relabeled from a recording file to a program file so the background task need not worry about allocating more Free Space. Program files are those in the directory that were previously recorded.

Hidden, Protected, Other…

Many additional attributes can be associated with a file depending on the sophistication of the desired product. For example, it would not be a significant increase in complexity to build a VOD Server that allowed each user to have a password protected file list. A large number of users could have a private view into the common list of all recorded files.

File System Tools

No matter how robust the software implementation, disk sectors will sometimes get lost in this type of application. There will be power failures and people will sometimes unplug the box. Occasionally losing a block of the disk cannot be avoided. Development must be allocated to create tools to find lost disk sectors, identify duplicates, de-fragment a heavily used disk, etc. Creating a large selection of file system diagnostic tools to help with developing and debugging of the system will be time well spent. Designing and implementing a file system with the intent of storing hundreds of gigabyte files that stream data at several megabits per second in an embedded environment is not for the faint of heart. Unfortunately, as mentioned above, most embedded operating systems are currently far from this level of sophistication. You will likely find that successfully meeting the challenge of developing a robust embedded multimedia file system will give your product a proprietary advantage for years to come.

User Interface

Certainly the type of product described here must have a fairly sophisticated user interface. Non-DVR/VOD details of this UI are well beyond the scope of this paper. The following are issues of particular interest to a DVR/VOD application intended to present multimedia both locally on a TV and remotely via a network connection.

Local Control & Display

IR Remote

For this product, the ubiquitous IR remote may be modified to include typical VCR (pause, fast forward, rewind, etc.) keys. For a multi-function remote, it would be good if these keys commanded the entertainment system's VCR from all modes other than "settop" mode. In settop mode, the VCR keys would talk to the settop and control the hard disk based pause/play modes instead of the VCR.

Front Panel

Simple input can also be received from front panel keys. However, it is tedious and difficult to operate from the front panel a standard digital settop that is receiving hundreds of channels. Trying to control DVR operation from a simple front panel seems impractical. It would probably be better to surrender and turn the front panel into a stylish smooth panel. If the remote gets lost, find it.

TV Screen Output

It is important to recognize that the TV screen is shared for both viewing of the program video and the presentation of the user interface. Many limitations are inherent with the use of the TV screen. There are issues with the rather poor resolution, interlaced display flicker, adjacent color bleed, and more. The DVR/VOD product described here offers the opportunity for unique features like automatically pausing the program when a full screen user interface is active. However, it will take clever usage of the TV screen to give the user convenient access to the many features offered by a DVR/VOD product.

Program Guide

Many consumer digital settop boxes have some type of a program guide to allow users to select a program for viewing. Our interest here is to highlight that there needs to be two more or less distinct types of program guides. One is a "standard" guide of current and upcoming programming available for viewing and/or recording. The other is a directory of previously recorded programs. For a consumer product of this nature it would be desirable to provide some form of parental controls. It may even be desirable to provide password protected user rights and hidden files. Finally, sophisticated program guide search features would be necessary to allow the user to track down and record desired programs.

Remote (LAN/WAN) Control & Display

Certainly it would be "cool" to connect to your home DVR/VOD device from work and command it to record a program in your absence. Why not be able to connect to it and command it while you are on travel several time zones away? In fact, why not be able to dump a few movies from it to your laptop so you can be entertained while you are flying to your destination? Given a solid fundamental architecture all this is possible (but perhaps not legal…). In fact, the more complex operation of a VOD Server in an apartment complex or large house benefits from a network connection to a sophisticated "browser" style user interface. Drag media and drop it on a destination or double click on your favorite movie to view it. A status bar can tell you how much Free Space is left on the disk. At a glance you can see the amount of network bandwidth being used and the number of active movies. From your browser (with some imagination) you can connect to your service provider's web site on "movie request night" and have your special request selected for broadcast. You can edit out commercials, collect all episodes of your favorite program, and insert yourself in your favorite movie. All these things are technically possible, but there is one unresolved issue…

Content Protection

Content protection is the hobgoblin of modern consumer electronics. There is no question that artists, distributors, and retailers need to be compensated for their work. Unfortunately there is significant disagreement on the details. There is even disagreement on the rights that consumers should ultimately have. Digital recording of media and the ease with which it can be distributed has created an enormous threat to the status quo. There are legal issues, moral issues, and tremendous uncertainty. Major corporations that both own content and sell consumer electronics products are schizophrenic. Content protection technology increases the cost of a product and may make it harder to use or less desirable to own. There is no question that the current climate is keeping some innovation out of consumer products, but there is some reason to believe that such limitations are righteous and moral. Content protection has all the ingredients necessary for well intentioned people to disagree for a long time.

Conclusions

Previous attempts at Video-On-Demand mostly failed because the great attraction of the service was the feature that made it infeasible. The diverse interests of customers combined with access to a large number of obscure titles consumed economically unsustainable amounts of bandwidth. But perhaps the dawn of cost effective and practical Video-On-Demand is at hand. As of this writing it is possible to get half a terabyte of hard drive storage for a few hundred dollars. Bigger and bigger hard drives combined with ever improving digital compression will soon yield platforms and services only dreamed of ten years ago. The dream of instant access to every movie ever recorded and all music ever made remains a dream for now, but instant access to hundreds of movies aggregated the previous month is a step in the right direction. Soon we me have hundreds or thousands of movies specifically selected or perhaps simply not rejected - sitting on your settop waiting for your viewing. Always there, always instantly ready