DISK-BASED VIDEO-ON-DEMAND STORAGE SERVERS: REQUIREMENTS, CHALLENGES AND (SOME) SOLUTIONS*

Yitzhak Birk
Technion - Israel Institute of Technology
Haifa 32000, Israel
birk@ee.technion.ac.il

ABSTRACT

A video-on-demand (VOD) storage server is a parallel, storage-centric system used for playing a large number of relatively slow streams of compressed digitized video and audio concurrently. Data is read from disks in relatively large chunks, and is then "streamed" out onto a distribution network. The primary design goal is to maximize the ratio of the number of concurrent streams to system cost while guaranteeing glitch-free operation. This paper characterizes the VOD applications and then identifies several important issues along with an outline of possible approaches to dealing with them. Issues include the accommodation of unlimited demand for concurrent "private" viewing of the few hottest movies with limited resources, multi-zone recording and the resulting variable disk transfer rate, as well the interplay between fault-tolerance, load balancing, the size of RAM buffers and the organization of the storage subsystem.

1. INTRODUCTION

1.1 Background

Video-on-demand, VOD for short, refers to a system and service that enable a very large number of end users to concurrently access large repositories of stored data, often of a stream nature such as video and audio, navigate through the material, choose items for viewing, and view them immediately. It is furthermore expected that the "feel" of the service would be one of a private repository. Important applications include movie libraries, educational and training material, video clips in for various applications, home shopping, personalized television programming, and probably many applications that have yet to be conceived.

When combining the required resources per active user with the expected number of concurrent users, this is perhaps one of the greatest challenges to computer and communication systems and their designers, as well as to the potential service providers. Presently, virtually all major computer and communications system vendors and service providers are engaged in research, development and initial deployment of VOD systems.

A system capable of providing VOD services comprises three major components: a video server, which is the main subject of this paper, user-premise equipment (sometimes referred to as a "set-top box"), and a distribution network. In large-scale, multivendor environments, various gateways are required as well. Each component comprises hardware as well as software. A discussion such heterogeneous environments appears in [1]2[3].

1.2 VOD servers

A VOD server comprises a storage subsystem, typically using magnetic disk drives as the primary storage device, a large RAM buffer, a streaming and network interface unit, an internal communication subsystem, and a control unit. Data is typically read from disk into the RAM buffer in relatively large chunks (in order to reduce disk-access overhead), and data for multiple video streams is then streamed out onto the distribution network in small units, such as ATM cells. While in the server, data may also be operated upon for purposes such as error correction, encryption and content-customization. Interesting papers on various issues pertaining to real VOD servers include [4] and [5].

VOD applications call for large systems and, unlike in many other applications, the storage subsystem plays a central role, not merely occupying a low level in the memory hierarchy. Also, most of a VOD server's cost lies in its storage subsystem. This warrants a careful look at the design of the storage subsystem for VOD. We next characterize the requirements placed on the server's storage subsystem. It should be mentioned that the design of the data paths within a server and the implementation of the streaming function are also challenging and the nature of the application permits unique solutions [6], but this issue is beyond the scope of this paper.

Storage subsystem requirements

A VOD storage server must provide a large number of concurrent streams of data. Each such stream is typically read from contiguous locations on one or more disks, and the rate of each stream is several times lower than the sustained transfer rate of a single magnetic disk drive. (1.5-6.0Mb/s/stream vs. 25-50 Mb/s/drive.) Once its viewing begins, a stream must not overrun or starve the available RAM buffers. High availability is also important. The server must respond promptly to user requests, but the response time to subsequent requests for data may be masked at the

* The work was supported in part by Hewlett Packard and by the Franz Ollendorff foundation. The author holds the Milton and Lilian Edwards Academic Lectureship.

cost of extra memory. Finally, the requirement for prolonged, glitch-free viewing implies that the tails of the probability distributions of various performance measures and resource use are of utmost importance, rather than simply their means or standard deviations.

VOD applications thus differ quite significantly from other prominent applications of large-scale storage subsystems: in on-line transaction processing, for example, performance is measured in accesses per second, and jitter is hardly an issue; in scientific computing, one often wishes to maximize the transfer rate for a single stream; in file servers, there is usually no notion of streams, and the exact performance measures depend on file size and the type of access. Much work has been devoted to optimization of storage performance in various applications. For example, the organization of data in a disk array used for OLTP is discussed in [7]. For general-purpose workstations, schemes such as placing the most latency-critical data in centrally-located tracks and placing different types of data in different disk drives have been proposed [8] [9]. VOD servers are have lately been receiving much attention, but much has yet to be done.

Storage-subsystem cost in a VOD server.

The cost of disk drives is the largest component of a VOD server's cost, and servers are expected to be bandwidth- rather than storage-limited, so disks must be used efficiently. However, one must also keep the cost of RAM buffers, which are required for masking disk response time and for storing the chunks of data received from disk, in check. (Data caching, and even reading ahead, are largely useless and even harmful in VOD servers.)

In the remainder of this paper, we present several examples of challenging problems that arise in video servers, along with possible solutions. Section 2 addresses the challenge of providing private viewing of a hot movie to an unlimited number of users with limited system resources, and offers a solution at an operational level. Section 3 discusses the ramifications of using disks with multi-zone recording and offers solutions at the level of intra-disk data placement. Section 4 discusses load-balancing among disks and fault-tolerance, and explores solutions in the form of inter-disk data placement and retrieval scheduling. Section 5 offers concluding remarks.

2. UNLIMITED PRIVATE VIEWING OF HOT MOVIES

New, "hot" movies are an extremely important source of revenue for service providers, who can even charge a premium during the first nights. Unfortunately, both theaters and tape-rental stores are unable to satisfy the demand. Can a video server provide the viewing flexibility of a rented tape to an unlimited number of viewers? This is clearly possible, but a brute force approach would be extremely costly. The challenge is to achieve this in a manner that does not require the system to be truly designed for such peaks, since the number of concurrent streams in this situation may be substantially higher than the normal aggregate load. In meeting the challenge, loads on the storage and control subsystems as well as the distribution network must all be considered.

Unlike interactive VOD applications, the viewing of feature films is passive. For "private" viewing, it suffices to permit the viewer to begin viewing at will, pause and resume at any time, and perhaps browse for the sole purpose of locating an interesting scene. Moreover, since a movie lasts nearly two hours and likely interruptions (due to phone calls and such) are on the order of minutes, resumption may be delayed by up to a minute or two. We next describe two solutions and evaluate them in terms of features and cost.

Hot movie in RAM

Placing a copy of an entire hot movie in RAM would completely remove any stress from the storage subsystem, and would allow complete viewing flexibility. The cost of RAM (1.5-6.0 GB for a feature film compressed using MPEG), compared with approximately 500KB per stream in a disk-based approach, would be prohibitive unless the number of streams is sufficiently large and the memory bandwidth is such that a sufficient number of streams can be played from the same physical memory. In any case, there would be no savings in the load on the control system, the streamer and the distribution network.

Staggered streams

Here, a "copy" of the movie is started at regular time intervals, say every two minutes. A viewer begins viewing a movie from the next new stream, within two minutes of his request. Requests to pause are granted instantaneously, and resumption is instantaneous from the "nearest" stream, within two minutes at the exact frame location, or some compromise between these two.

The load on the storage system is merely 50 streams (assuming a 100 minute movie). Thus, this scheme is already beneficial with a moderately large number of viewers, and becomes more so as the number increases.

This scheme can easily take advantage of the multicast nature of distribution networks such as cable TV. By taking up some 50 out of more than 750 possible streams, the network's capacity for other purposes is only mildly affected. (Of course, the situation would become worse if more than one movie were shown in this way, but would still be the most efficient once there are at least 50 viewers per movie.) For distribution networks based on point-to-point links, the scheme would require the same link resources as true private viewing, but may have advantages if the switching system is capable of multicast.

In a multicast distribution network, all streams are present at the input of the user-premise equipment. It is therefore possible to implement all the private-viewing functions locally: at setup time, the server would tell the user-premise equipment which channels are being used and the starting times; the set-top box would then simply translate the viewer requests into a choice of channel.

In view of the above, the staggered streams scheme appears to be most attractive. Staggered streams could also be implemented with the entire movie in DRAM, but the 50 fold increase in RAM use in return for reducing the load on the storage subsystem by an amount equivalent to the streaming capacity of three disk drives would not be justified.

In summary, this extremely important problem can be solved at a policy level.

3. MULTI-ZONE RECORDING AND VOD SERVERS

Modem magnetic disk drives employ multi-zone recording, which is a close approximation of fixed linear recording density: the disk is partitioned into sets of contiguous tracks, called zones, and track capacity within each zone is equal to the maximum permissible capacity of the zone's innermost track. Such disks rotate at fixed RPM, so transfer rate depends on track location. A typical dynamic range can be as high as 1:1.8 [10][11].

If a movie occupied a large fraction of a disk, the permissible number of concurrent viewers of any given movie would change with time. If, as is often the case, each movie is striped across several disks and occupies only (the same) small fraction of each of them, the permissible number of viewers only changes when they make their selections, but does depend on those. Since the number of concurrent video streams is the most important measure of a VOD server's performance, this issue must be addressed.

One approach is to try and exploit the track-dependent transfer rate by placing the most frequently viewed movies in the outermost tracks, and the least frequently viewed ones in the innermost tracks. This approach, referred to as *load matching*, maximizes the expected value of the permissible number of concurrent streams. However, the viewing pattern may deviate significantly from the "typical" one as it changes with the time of day or in response to unexpected events. In this case, the permissible number of streams may drop by tens of percents, and the problem may persist for an hour or so. One could dynamically rearrange the material on disk, but this takes up time and bandwidth, especially in fault-tolerant systems.

An alternative "static" approach, referred to *as load balancing*, is to maximize the *guaranteed* (over the range of viewing choices) permissible number of concurrent video streams. This can be done using randomized layouts but it is then difficult to guarantee the short-term behavior. There are, however, two deterministic schemes for achieving this goal.

Logical Tracks [12]

For a disk that has like numbers of tracks in all zones, one constructs fixed-size logical tracks comprising an equal number of same-numbered physical tracks from every zone. (See Fig. 1.) While the original purpose of this scheme was apparently to adapt multi-zone disks for use with operating systems that can only handle fixed-size tracks, recording each movie in logical-track order would guarantee sustained transfer rate at playback time which is independent of viewing choices.

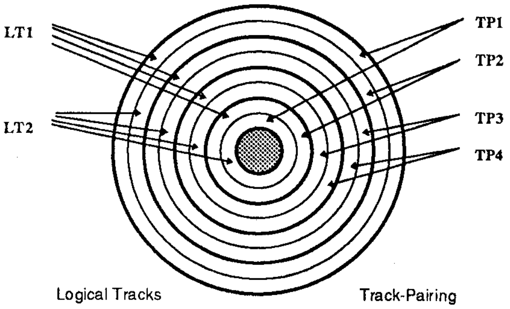

Fig. 1 A representative surface of a disk drive with four recording zones, each with two tracks. An arrangement into two logical tracks is sown on the left, and into four track-pairs - on the right.

Track-Pairing [13]

This scheme is based on the observation that with fixed linear density, track capacities form an arithmetic sequence. By conceptually pairing the innermost track with the outermost one, the second innermost with the second outermost, etc., both the capacity and the net reading time are the same for all pairs, and consequently so is the transfer rate. (See Fig. 1.) By recording a movie alternately on a range of contiguous "outer" tracks and their "inner" counterparts, the disk's throughput becomes independent of viewing choices with essentially no penalty in terms of disk overhead. The method has been implemented on an HP C2247 1GB disk drive under the Microsoft Windows NT operating system [14], and implementation on a pair of IBM disk drives (pairing track *I* on one disk with *N-i* on the other) is nearing completion. The method has also been extended to multiple disks and disks with multiple arms. Finally, we note that Track-Pairing can be combined with load matching by excluding a band of tracks from the pairing and reserving them for hot movies.

Forming (exact) fixed-length logical tracks with Track-Pairing would require complicated bookkeeping, but this is not necessary for VOD applications. On the other hand, this scheme has important advantages over logical tracks in terms the buffers required to mask the short-term variability in transfer rate, since the fixed rate is achieved after two chunks. The advantage is even more pronounced when used in conjunction with error-correcting arrays in the presence of a faulty disk drive. For a detailed description of Track-Pairing and a comparison with Logical Tracks, see [13].

4. RAIDs AND THE BUFFER-SIZE EXPLOSION

In the case of partitioning data among disks, unlike that of placing the data within a disk, there is no tradeoff between load balancing and load matching: under both measures, the optimal partitioning entails striping the data for every movie across all disk drives. The granularity of the striping is determined by weighing disk utilization, which is maximized by coarse striping, against RAM buffer size which is minimized with fine striping. Since there is no sense in making the granularity of the data placement coarser than one chunk (a chunk is the amount of data read from a single disk drive in a single time slice), a reasonable size would again be on the order of 128-256KB.

Disk drives are very reliable devices, with a calculated MTBF of nearly one million hours. Consequently, even in a system with hundreds of disk drives, the failure rate may be acceptable. Moreover, the data is mostly prerecorded, so one can keep a spare copy on tape and there is no fear of losing data. However, since data for any given movie is striped across many, possibly all, disk drives, any failure constitutes a common event to numerous users. (Consider, by analogy, a city-wide power blackout twice a year versus a random light bulb burning out once a week. Clearly, the former is highly undesirable, whereas the latter is acceptable.) Consequently, high availability is of utmost importance.

One could try to provide high availability by keeping a spare, empty disk drive, and loading it with the data that used to be on the drive that failed. Doing so by keeping each movie on a tape and rebuilding the disk from tapes would be prohibitively time-consuming, since numerous tapes would have to be loaded and unloaded, and entire tapes would have to be scanned since the appropriate data is not contiguous. Alternatively, one could keep an image of every disk on tape; the usefulness of such a scheme would depend on the frequency of writing to the disks.

Another option is to use redundancy that permits reconstruction of the faulty disk's data from the data on other disks. A disk array that employs such techniques is known as RAID [15].

The conventional use of RAIDs, either at all times or whenever a disk has failed, entails reading an entire stripe into memory and reconstructing the data of the faulty disk if there is one. Once the data is used for reconstruction, it can be discarded and read again when needed at the cost of a doubling in storage bandwidth; this is unacceptable. Alternatively, it can be stored until needed for playing, but this is also unacceptable for VOD applications and large arrays, as explained below.

The size of the chunk of data read contiguously from a single disk is chosen based on disk-utilization considerations, and is independent of the array's size. Therefore, the amount of data read on behalf of a stream increases linearly with the size of the array across which it is striped, and so does the amount of RAM buffer required for each stream. Since the maximum number of streams also increases linearly with the number of disks, total RAM size would increase quadratically with system size. (In other applications, the data read for reconstruction is normally either of immediate use to the processor or will not be used and can be discarded. In VOD applications, this data is useful, but not immediately, making buffering very expensive.)

One way to overcome this problem is to stagger the access schedules to the different disks, so each stream is served by at most one disk at any instant. This requires only a constant buffer size per stream. Fig. 2 depicts the RAM buffer occupancy for a single stream as a function of time with simultaneous access and staggered access. With staggered access, however, the parity group is unavailable for reconstruction, and making it available would again require the large amount of memory.

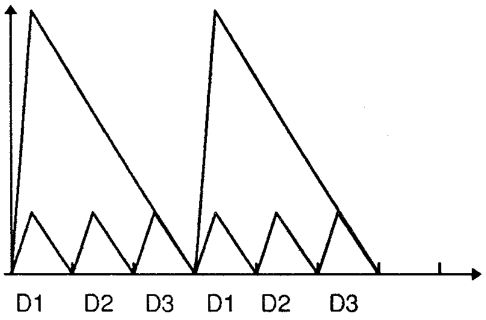

Fig. 2 Occupied buffer space vs. time, shown for a single stream in a 3-disk array. The "fine teeth" represent staggered access and the "coarse" ones represent simultaneous access to all disks.

A possible compromise is to partition the disks into arrays of fixed size, say $k+1$. (This increases the storage overhead from $1/M$ to $1/k$.) Each array would be operated as a conventional RAID, but the access schedules to the different arrays would be staggered. For a system with M disks, the total RAM buffer size would increase as $k \bullet M$, which is linear in M. We have been examining other alternatives, which show very promising results in simulations. For example, it appears that we can run a system of some 30 disks at over 90 percent of theoretical streaming capacity with a storage overhead of at most 20 percent, a small buffer size per stream (a small number of chunks), and other overheads also held down to low values. Our research is continuing in various directions, including examination of the dynamic behavior of the system.

A final option is to keep two copies of frequently-viewed movies, and rely on one of the previous schemes for reconstruction of infrequently viewed ones. The quality of service offered by such a scheme and its cost depend on the viewing-frequency distribution: if very few movies attract nearly all the viewing, the scheme is excellent. At the other extreme, the scheme is very costly or offers poor service if all movies are equiprobable.

5. CONCLUSION

The cost-effective provision of VOD services to the masses is extremely challenging, and presents the various subsystems with a set of requirements that often differ from those presented to them by other applications. In this paper, we touched upon several issues, and demonstrated that solutions may be found at different levels of system design and operation. Much further research is needed, and new challenges are likely to arise once large systems are deployed.

Acknowledgment. The donations of disk drives by IBM, software by Microsoft and a PC by Intel are gratefully acknowledged.

REFERENCES

[1] S.D. Dukes, "Next generation cable network architectures," Proc. NCTA Conf., pp. 8-29, 1993

[2] W.D. sincoskie, "System architecture for a large scale video on demand service," *Comp. Net. and ISDN Sys.*, vol. 22, pp.155-162, North-Holland, 1991.

[3] P.V. Rangan and S. Ramanathan, "Designing an on-demand multimedia service," *IEEE Commun., July 1992.*

[4] F.A. Tobagi and J. Pang, "StarWorks - a video applications server," IEEE 1063-6390/93, pp. 4-11, 1993.

[5] R.L. haskin, "The Shark continuous-media file server," IEEE 1063-6390/93, pp. 12-15, 1993.

[6] Y. Birk and R. Perets, "A switch for large-scale video servers," work in progress.

[7] J. Gray, R. Horst and M. Walker, "Parity striping of disc arrays: low-cost reliable storage with acceptable throughput," Proc. 16th Intnl. Conf. on Very Large Databases, Brisbane, Australia, pp. 148-159, Aug. 1990.

[8] C. Ruemmler, J. Wilkes, "Disk shuffling," Hewlett Packard Technical report HPL-91-156, Oct. 1991.

[9] K. Muller and J. Pasquale, "A high-performance multi-structured file system design," Proc. 13th ACM Symp. on Operating System Principles (SOSP), Asilomar, CA, October 1991, pp. 56--67.

[10] Hewlett Packard Company, C2240 SCSI-2 disk drive - Technical Reference Manual, 2nd ed., P/N 5960-8346, April 1992.

[11] Hewlett Packard Company, C2486A/88A/90A SCSI-2 Disk Drives -- Technical Reference Manual, 1st ed., Sep. 1992.

[12] S.R. Heltzer, J.M. Menon and M.F. Mitoma, "Logical data tracks extending among a plurality of zones of physical tracks of one or more disk devices," U.S. Patent No. 5,202,799, April 1993.

[13] Y. Birk, "Track-Pairing: a novel data layout for VOD storage servers," Hewlett Packard Technical report HPL-95-xxx, to appear, 1995.

[14] E. Almog, N. Kogan and I. Tadmor, "Video file server prototype," project report, Parallel Sys. Lab, EE Dept., Technion, Haifa, Israel, May 1994.

[15] D.A. Patterson, G. Gibson and R.H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," Proc. ACM SIGMOD, pp. 109--116, June 1988.