CMPSCI 754 Multimedia Systems **Spring 2001**

Lecture 7: 22 Feb '01

Lecturer: Prashant Shenoy *Scribe: Saif Raza Hasan*

In this lecture we shall look at the possibility of having a variable block sizes and how that affects the design of the storage system.

We shall also discuss fault tolerance techniques in media servers.

Variable Block Sizes

In a system with variable block sizes the assumption is that the number of frames per block is the fixed. Therefore the size of the blocks depends on the size of the fixed number of frames.

Since the frames are going to be accessed with a round based strategy and a fixed number of frames are requested in each round, the same number of blocks will be retrieved in each round. Therefore the load will move in a lock-step manner. This kind of periodic behavior of load is not found in Fixed Block Size because each block may have a different number of frames.

Therefore to minimize the seek and rotational latency all the frames which are accessed in a round should be stored in a single block. Therefore each client will access will exactly one block in each round and an increase in the block size will increase the round duration and vice versa.

Optimal Media Block Size

The factors that affect the choice of the block size are:

- load balance
- latency overhead

But since in this case, we have a definite disk access pattern, we can make a client wait so as to synchronize it's access with the ongoing rounds. In this manner it is possible to ensure that the load on each disk is the same. Therefore the load balancing criterion is not relevant in this case.

Therefore to minimize the latency the block should be made as large possible (in fact exactly equal to the size of one round).

The trade-offs involved with this are:

- Increased Buffer Requirement
- Increased initiation latency (because of synchronizing access with on going rounds.

Comparison of Fixed-size and Variable-size Blocks

The comparison of Fixes and Variable sized blocks can be summarized as below:

- Fixed Size Blocks
  - Advantages
    - The management of storage space is simple.
  - Disadvantages
    - Admission control of clients is complicated.
    - It is harder to do load balancing
- Variable Size Blocks
  - Advantages
    - The retrieval of the video streams happens in a lock-step, therefore it is possible to ensure load balancing at the time of admission.
  - Disadvantages
    - Because of the different size of different blocks, there is an inherent complexity in the allocating and de-allocating of blocks.
    - There is a higher implementation overhead.

Effect of Degree of Striping

There are two kinds of striping across disks possible.

Wide Striping In this case, the blocks are striped across all the disks in an array.

- This approach distributes the load across all the disks in the array.
- It also prevents the occurrences of 'Hot Spots' (A single or a group of disks getting overloaded)

Narrow Striping In this case the stripe is across only a subset of the disks in the array. Therefore, the degree of striping is strictly less that the no. of disks. For Narrow Striping, the array has to be partitioned and the files on the server have to be mapped to the various in such a manner so that the load across the partitions is balanced. The partitions may or may not overlap each other.

The figure on lecture slide 71 illustrates the performance of wide striping for increased disk numbers and number of users supported. The actual performance is lower than the expected behavior. This can be explained on the following basis:

1. For a fixed load, increasing the number of disks for striping increases the imbalance because of sparse loading.
2. But increasing the load decreases the imbalance.

An increase in load imbalance decreases the utilization and if the optimal block size decreases the throughput decreases.

However the increase in imbalance due to (1) above dominates the decrease due to (2) and hence we have the departure from the expected linear behavior.

Therefore to obtain an optimal degree of striping, the following approach is taken:

- Partition the array and stripe each file across a sub-array
- Allocate files to partitions in such a way that each partition is equally likely to be accessed.

The tradeoffs involved in such a design are:

	Imbalance within partitions	Imbalance across partitions
Small Partitions Size	Small	Large
Large Partitions Size	Large	Small

The partition size should be chosen to balance these tradeoffs. The two parameters used for this determination are:

- The number of clients supported within a partition
- The number of clients supported by the entire array

The next step is to determine an efficient mapping of files to the partitions. The objective again is to balance the load across the partitions. To be able to do this effectively we assume that the access frequency of objects is known.

Based on this information the files should be assigned to partitions such that the total access frequency for all partitions is identical. Frequently accessed files can be replicated on different partitions. Dynamic load balancing and replication may be necessary.

Fault-Tolerance Techniques

As the number of disks in a system are increased the frequency of disks failures also increases. E.g. The typical MTTF (time to failure) for conventional disks $\approx$ 200,000 hours. This implies that a multimedia server consisting of 500 disks will see a failure every 400 hours.

The objectives of a failure recovery scheme should be to:

- Quickly rebuild information stored on the failed disk, without taking the system off-line.
- Continue to meet the real-time performance requirements of video streams during failure recovery.

Two alternative approaches to failure recovery in disk arrays are:

Mirroring This involves completely duplicating all data on the disks. This has 100 % storage space overhead.

Parity Encoding This scheme uses data recovery using redundant parity information. RAIDs (Redundant Arrays of Inexpensive Disks) have several different schemes.

RAID-3 uses Bit-interleaved parity for recovery.

RAID-4 The parity information is maintained on a dedicated disk. However in this case the parity disk becomes a bottleneck

RAID-5 The RAID-5 scheme tries to avoid hot-spot contention by scattering parity information across the array.
However the disadvantage of this scheme is that if the system load is balanced prior to failure, then the surviving disks would observe a 100% increase in load.
Also preventing saturation gives lower utilization in the fault-free state

De-clustered Parity Architectures

This approach trades storage space for improved performance. The idea is to protect a smaller number of data blocks (say (G-1)) using each parity block and to distribute the parity information across all disks.

In this case, the increase in load experienced by the surviving disks in the case of failure = $\frac{(G-1)}{(D-1)}$ where D is the number of Disks.