Lecture 11: Mar 13

Lecturer: Prashant Shenoy *Scribe: Ping Ji*

Admission Control and Caching

11.1 Retrieval Techniques Overview

The objective of Retrieval Techniques is to maximize the number of clients that can be serviced simultaneously. The Retrieval Techniques include employing disk scheduling algorithms to efficiently access blocks and reserving server resources to ensure that the real-time playback requirements of clients are met. In next section, we'll talk about reserving server resource.

11.2 Resource Reservation

The goal of resource researvation is to limit the total service time for all of the clients. Resource reservation techniques involve estimating the resource requirement of a new client, estimating the total resource requirement assuming the new client is admitted, and admitting a new client only if the total resource requirements is less than the capacity. Which criteria to be used for the admission control depends on the service requirement of the clients.

11.2.1 Service Requirements

The clients of a multimedia server can be seperated as either *tolerant* or *intolerant* clients.

- **Tolerant Clients:** can tolerate brief distortions and play back discontinuity. For this kind of clients, we need to specify what can be tolerated by them.
- **Intolerant Clients:** require deterministic service guarantees

The tolerancy of a client can be used to classify an admission control algorithm.

- **Deterministic algorithms:** The deterministic admission control algorithm provide strict guarantees. It can be used for intolerent client due to their property of being restrictly deterministic.
- **Statistical algorithms:** The statistical algorithms provide statistical guarantees for the service. It means that the probability distributions are used to make assumption of whether the system can meet the requirement of a client. Based on the probability, server controls the adimission requirements, the service is probabily guaranteed.
- **Predictive algorithms:** The predictive algorithms provide acceptable performance but without absolute guarantee. It can be used to serve those tolerant clients. But for those intolerant clients, it's not appropriate.

For above three classes of admission control algorithms, we notice that the status of service guarantee is getting weaker from top to down. We will explore the details of these three classes of algorithm in the following section.

11.2.2 Admission Control Algorithms

1. **Deterministic Admission Control**
   The **admission control criteria** is that, for the worst case, the total response time should not be greater than the maximum number of duration of a round, $\mathcal{R}$.

$$k * (t_s + t_r + t_x) \leq \mathcal{R} \tag{11.1}$$

   Where, k: maximum number of blocks accessed in a round
   t_s: worst case seeking time to access a block
   t_r: worst case rotational latency to access a block
   t_x: transfer time of a block

   Here, we notice that the deterministic admission control algorithm utilizes the worst case estimations, which is the case that for the total requirement the max disk space is exceeded, for the number of blocks to be accessed and their relative positions on the disk. The deterministic admission scheme ensures that continuous retrieval requirements of clients are strictly met. The process of the deterministic admission control follows steps, 1. get the request from a new client; 2. estimate the worst-case required; 3. evaluation based on equation (11.1); 4. make decision of accept or deney.

2. **Statistical Admission Control**
   The key concepts of the statistical admission control algorithms are to use probability distributions, rather than the worst-case values, to estimate the service times and number of blocks to be accessed from disks, and to provide statistical service guarantees to each client. In this kind of algorithm, the probability distribution of load is used to estimate how likely the disk can meet the requirement of all the client, and the time to access the blocks is vary based on the estimation.

   The **admission control criteria** is as follows,

$$E[\# \ of \ frames \ can \ be \ retrieved] = q * \mathcal{F}_0 + (1-q) * \sum_{i=1}^{n} f_i \geq \sum_{i=1}^{n} p_i * f_i \tag{11.2}$$

   Where, $\mathcal{F}_0$: is the minimum number of frames guaranteed to be accessed in a round. It depends on the disk.
   q: is the overflow probability where $q = P(\tau > \mathcal{R})$. And $(1-q)$ is the probability of no overflow. The service time τ may vary from one round to another due to variation in the number of blocks to be accessed during a round and the variation in their relative placement on disk;
   p_i: is the loss affordability of client i
   n: is the number of clients
   f_i: is the frame rate of each client

   The overflow probability q can be computed using,

$$q = P(\tau > \mathcal{R}) = \sum_{k=k_{min}}^{k_{max}} P(\tau_k > \mathcal{R}) P(\mathcal{B} = k) \tag{11.3}$$

   Where, τ_k : is the time to retrieve k blocks
   $\mathcal{B}$: is the number of blocks retrieved in a round
   $k_{min}\&k_{max}$: correspond to the smallest (or maximum) # of blocks can cause an overflow

Also the determination of minimum number of frames guaranteed to be accessed in a round, $\mathcal{F}_0$, is based on the determination of the minimum number of blocks (say k_d) that are guaranteed to be accessed during each round of duration $\mathcal{R}$, and the determination of the minimum number of frames that can be contained in k_d blocks.

3. **Predictive Admission Control**
The predictive admission control scheme says that if the measurements of the server load from the recent past indicate that the service requirements of clients can be met, then admit this new client.

The **admission control criteria** of predictive admission control algorithms is,

$$(\hat{k}_{new} + \sum \hat{k}_i) * \hat{\eta} \leq \mathcal{R} \tag{11.4}$$

Where, $\hat{\eta} = E(\eta) + \epsilon_1 \cdot \delta(\eta)$;
$\hat{k}_i = (E(k_i) + \epsilon_2 \cdot \delta(k_i)) * p_i$;
k_i: is the number of blocks accessed by client i;
η: is the time to retrieve a media block;
$\delta(x)$: is the standard deviation of x
The determination of $E(\eta)$ and $\delta(\eta)$ is trhough measurement, and the determination of $E(k_i)$ and $\delta(k_i)$ is from precomputed statistics

11.2.3 Overload Control

The schemes of admission control introduce a problem of spreading the losses to multiple clients. An aggressive admission control algorithm can imply occasional violations of continuity requirements (i.e. $\tau > \mathcal{R}$). The reason for it is the decision of dropping a block. If one block need to be dropped, then which one should be that being dropped one? By observation, sequential nature of video playback enables the server to predit overload. Thus before the block drop, we can predict the probability of the overflow first. The idea of overload control is introduced.

The objective of overload control is to minimize the number of blocks being discarded/delayed, and to distribute such blocks among tolerant clients. The *basic idea* of overload control is to select a set of blocks to be discarded/delayed based on the loss affordability of clients as well as the relative placement on disk. And discarding blocks should also maximizes the reduction in service time per discarded frame.

The **Overload Control Algorithm** is as following,

1. Compute the loss affordability of client i during round j as

$$\chi_{i,j} = L_{i,j} - l_{i,j} \tag{11.5}$$

where, $L_{i,j} = \lfloor j * (1 - p_i) * f_i \rfloor$ and $l_{i,j} = \sum_{m=1}^{j-1} \hat{l}_i^m$. And $\hat{l}_i^m$ denotes the number of frames of client i discarded during the mth round.

2. Using $\chi_{i,j}$ label each media block to be retrieved during the round as either *can-be-discarded* or *can-not-be-discarded*. When implementing this step, the number of blocks to be dropped should be minimized.

11.3 Techniques for Increasing Server Capacity / Video Caching

In conventional systems, cahcing is used to improve program performance (e.g. reduce transmission delay). and in video servers, the schemes of batching and caching are used to increase the server capacity. In this section, we'll introduce the basic ideas of Batching and go through the details of video caching.

11.3.1 Batching policies:

The *basic idea* of Batching is to group clients requesting the same video object that arrive within a short duration of time. That can be understood as that when the requests come, the system waits for some amount of time to gather a group of requests on the same object, batch those requests together and serves them at once.

The tradeoffs of this batching scheme are,

- Improvement the capacity of a server, because the remain bandwidth can be used to serve other requests
- Large initiation delay. The reneging probability is larger. Here, by reneging probability, we mean the probability of a user waits for a long time and lost interest even gave up.

11.3.2 Video Caching Schemes:

The caching technologies for streaming media is different to the technologies for caching text files. In video server, the main purpose of using caching teniques is to improve the capacity of a server, rather than to provide better performance. An ealier receiving of a file is not what we are most interested in by using video caching.

1. **Interval Caching:** Interval caching is to cache only the interval between two temporally spaced clients.

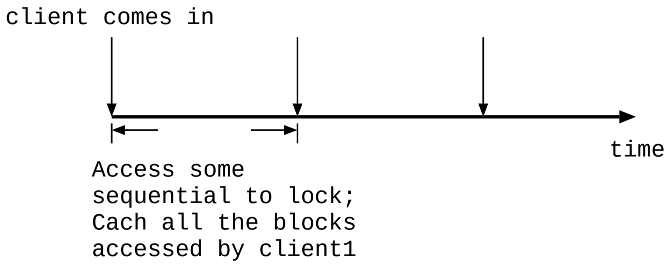

Figure 11.1: Interval Caching

In Figure 11.1, we can see that the interval caching trys to cache the interval of two clients accessing the same file. The buffer is accessed sequencially. When the caching buffer is full, there will be some blocks removed for making room for incoming requests. This introduces a problem that if we do not have enough buffer, how to decide which interval need to be cached? To explain this problem, we assume that the cache caching a subset of intervals, need to replace some cached intervals. Always discard larger intervals so that the released blocks can support more clients and provide larger capacity.

2. **Frequency Caching:**
   In frequency caching, replacements decision is based on the frequency. The most frequently accessed video objects will be cached. In this scheme, a large buffer space is required.

3. **Taxonomy of Cache Replacement Policies:**
   1. Recency of access: LRU, LRU-k, which make the decisions of which blocks should be discarded depending on the (k)most recent access.
   2. Frequency based: LFU, FBR
   3. optimal: TTR (Time-to-reacees)
   4. Size-based (LRU-MIN, GD-size), Miss cost based (GD-size), and Resource-based caching uses cost functions to make the discarding decision.