

CS 414 – Multimedia Systems Design
Lecture 30 –
Media Server (Part 4)

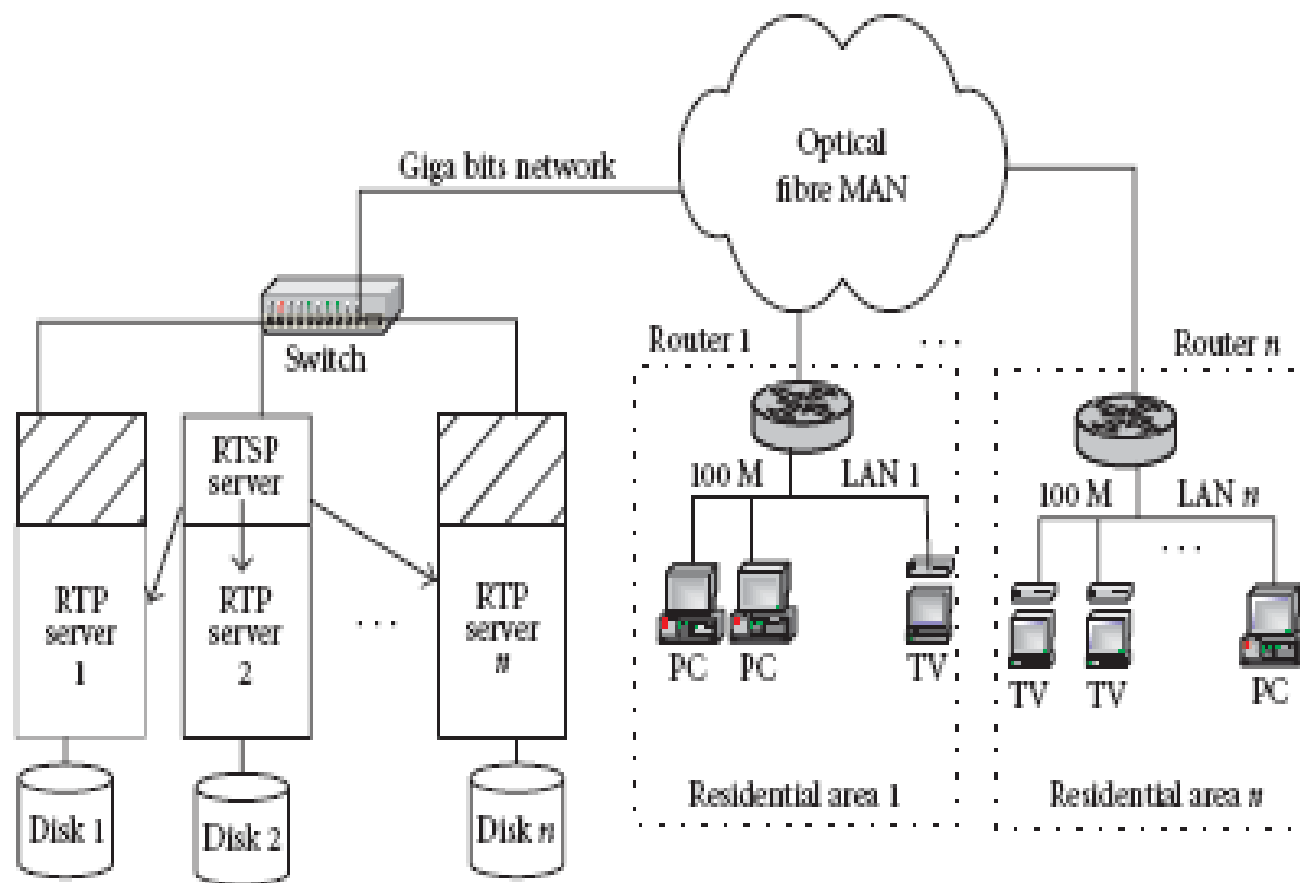
Klara Nahrstedt
Spring 2008



Outline

- Media Server Model
- Validation of Model
- Batching

Example of Media Server Architecture



Source: Medusa (Parallel Video Servers), Hai Jin, 2004



Factors affecting Optimal Block Size

■ Server Configuration

- Number of disks in the array
- Physical characteristics of disks
- Type of disk array (i.e., redundant vs. non-redundant)

■ Client Characteristics

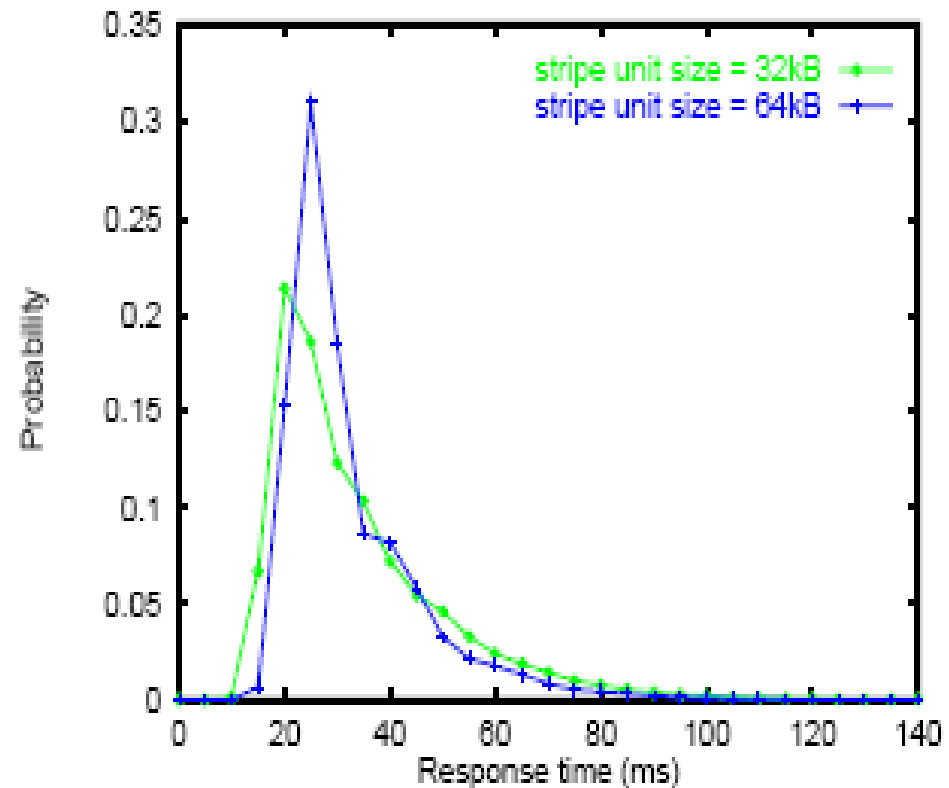
- Number of clients accessing the server
- Client request size

- **Objective:** given server configuration, client characterization, determine the optimal block size

Selecting Metrics

Text

- **Aperiodic accesses**
=> client-pull
architecture
- **Best effort** =>
minimize response
time

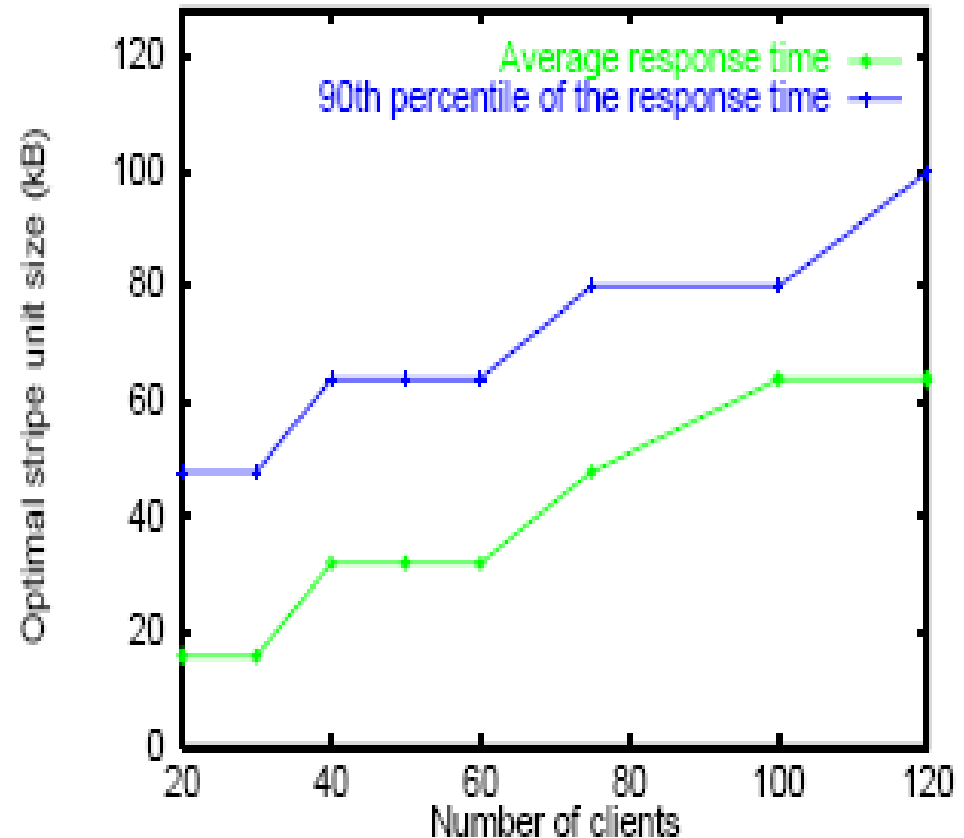


Graph Source: Shenoy Prashant, U. Mass., 2001

Selecting Metric

Continuous Media

- **Periodic and sequential access** => server-push architecture
- **Real-time** => minimize tail of response time distribution



Metric: Minimize service time of the most heavily loaded disk

Graph Source: Shenoy Prashant, U. Mass., 2001



Media Server Modeling

- Given: Server Configuration and Client Characteristics
- Objective: **predict service time of the most heavily loaded disk**
- Main steps:
 - Estimate number of blocks access from disk by single client
 - Estimate total number of blocks accessed from disk
 - Estimate number of blocks accessed from most heavily loaded disk
 - Compute the service time of most heavily loaded disk

Model Source: Shenoy Prashant, U. Mass, 2001

Model

- Use frame size distribution to determine the total number of blocks accessed (C_i) from the array by a client (let $P(C_i = k) = b_i^k$)
- Number of blocks accessed from disk j by client i (X_i^j)
 - Property: A request is equally likely to start on any of the D disks

$$\Rightarrow P(X_i^j = 1) = \frac{b_i^1}{D} + \frac{2 \cdot b_i^2}{D} + \dots + \frac{D \cdot b_i^D}{D}$$

$$P(X_i^j = k) = \sum_{m=1}^D b_i^{(k-1)D+m} \cdot \frac{m}{D} \quad (k = 1, 2, 3\dots)$$

- Total number of blocks accessed from disk j by n clients

$$Y^j = \sum_{i=1}^n X_i^j$$



Model

- Number of blocks accessed from the most heavily loaded disk

$$Y^{\max} = \max(Y^1, Y^2, \dots, Y^n)$$

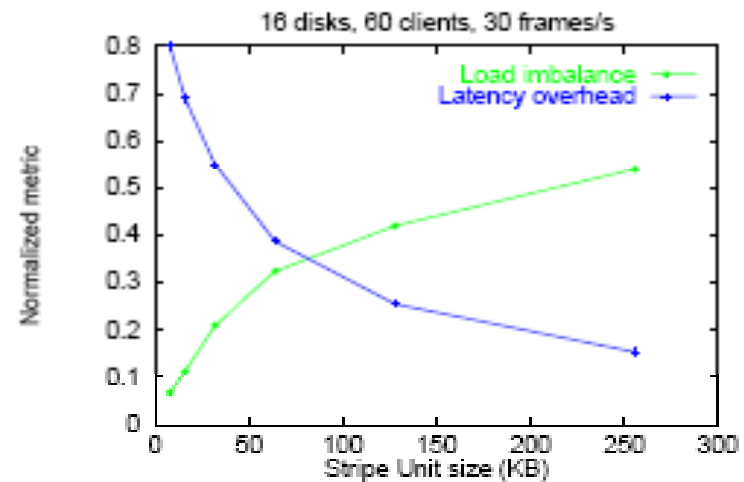
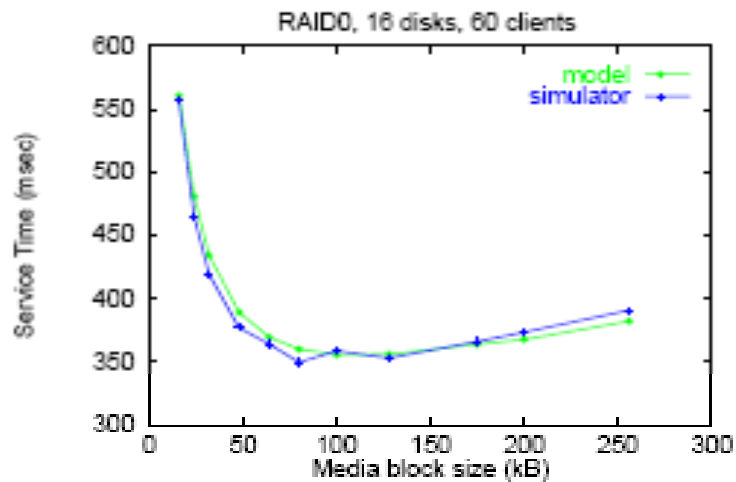
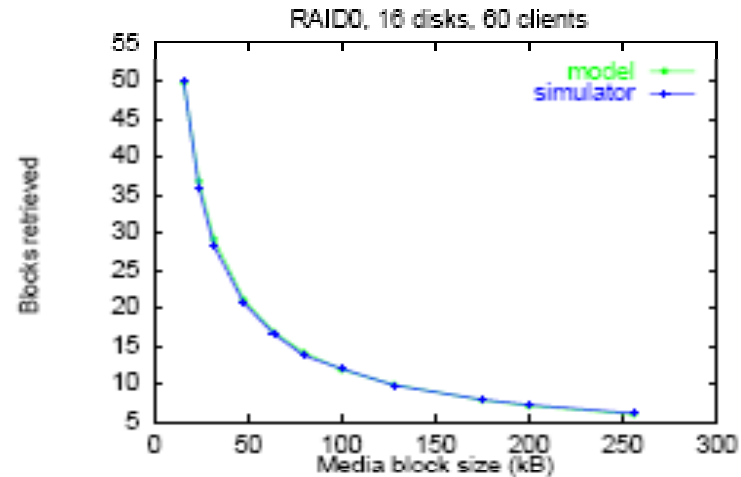
- Service time of the most heavily loaded disk

$$\tau = E(Y^{\max}) * (t_s + t_r + t_x)$$

where

- t_s : seek time to access a block
- t_r : rotational latency to access a block
- t_x : transfer time of a block

Validation of Model





Techniques for Increasing Server Capacity

- Batching
- Caching
 - Interval Caching
 - Frequency Caching
- Key Point
 - In conventional systems, caching used to improve program performance
 - In video servers, caching and batching are used to increase server capacity



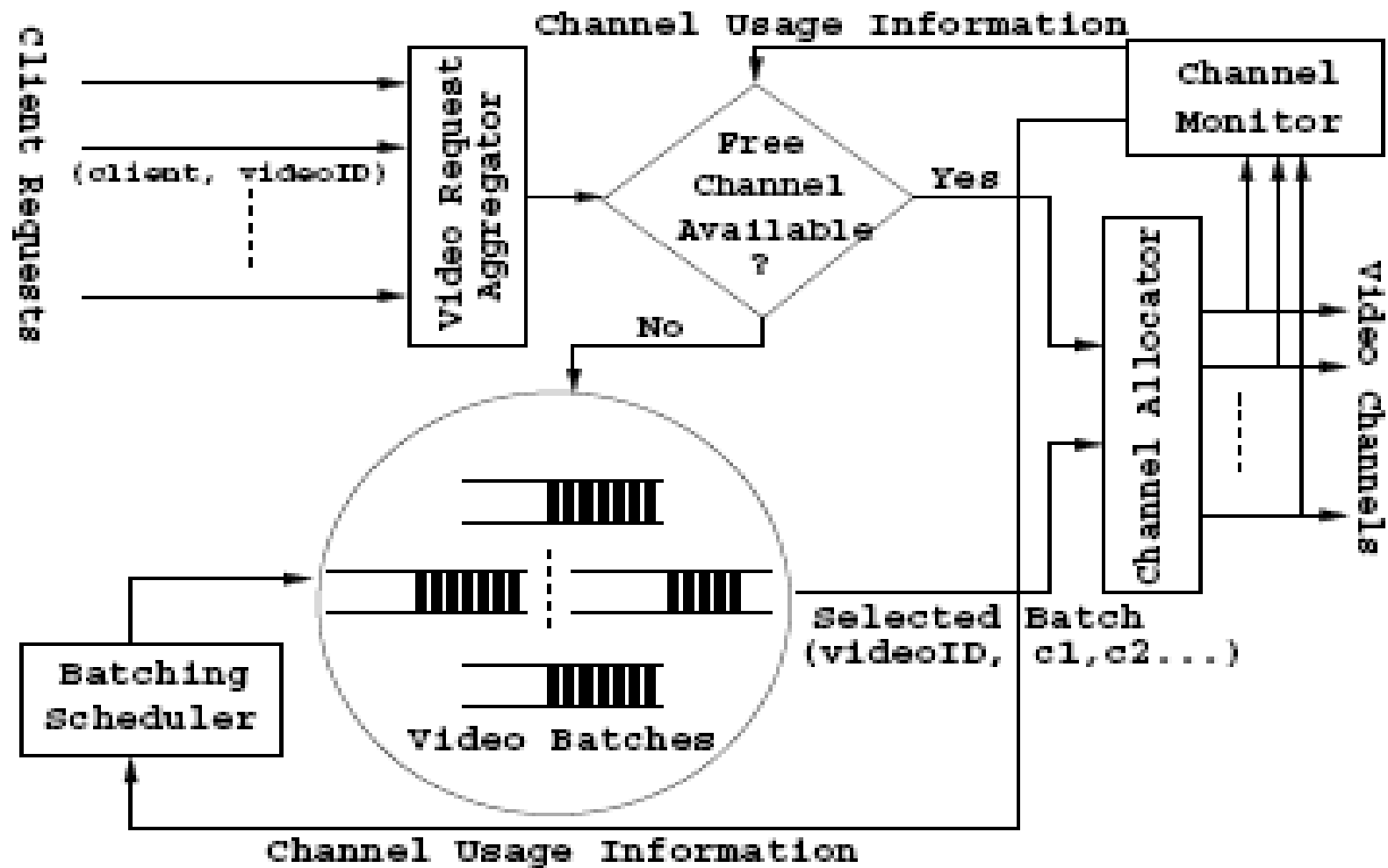
Batching

- Group clients requesting the same video object that arrive within a short duration of time or through adaptive piggy-backing
- Increasing batching window increases the number of clients being served simultaneously, but also increases reneging probability
 - *Increasing minimum wait time increases **client reneging***
- *Performance* metrics: latency, reneging probability and fairness
- Policies:
 - **FCFS**, **MQL** (Maximum Queue Length), **FCFS-n**



Batching Policies

- **FCFS:** schedules the batch whose first client comes earliest, with the aim of achieving some level of fairness
- **Maximum Queue Length:** schedules the batch with largest batch size, with the aim of maximizing throughput
- **FCFS- n :** schedule the playback of n most popular videos at predefined regular intervals and service the remaining in FCFS order



Source: "Selecting among Replicated Batching VOD Servers, Guo et al. 2002



Conclusion

- The data placement, scheduling, block size decisions, caching are very important for any media server design and implementation.