DISTRIBUTED MULTIMEDIA RETRIEVAL STRATEGIES FOR LARGE SCALE NETWORKED SYSTEMS
MULTIMEDIA SYSTEMS AND APPLICATIONS SERIES

Consulting Editor

Borko Furht
Florida Atlantic University

Recently Published Titles:


Visit the series on our website: www.springeronline.com
DISTRIBUTED MULTIMEDIA RETRIEVAL STRATEGIES FOR LARGE SCALE NETWORKED SYSTEMS
by Bharadwaj Veeravalli and Gerassimos Barlas

e-ISBN-10: 0-387-29675-1
Dedicated to:

My Parent’s (late),
My Wife Lavanya,
My Daughters Tanya & Harshitha,
All my Teachers who got me here!

—Bharadwaj Veeravalli

My wife Katerina
My parents, Dimitrios and Maria

—Gerassimos Barlas
Contents

List of Figures xi
List of Tables xix
Preface xxvii
Acknowledgments xxvii

1. DISTRIBUTED TECHNOLOGIES FOR MULTIMEDIA RETRIEVAL OVER NETWORKS 1
   1.1 Video on Demand: The Challenge and Contemporary Solutions 1
   1.2 General Multimedia Storage and Retrieval Architectures 3
      1.2.1 Multimedia servers: A resource management perspective 3
      1.2.2 Client-Server VoD system 6
      1.2.3 Designing an admission control algorithm: Single Server multiple clients system 7
      1.2.4 The Fellini Multimedia storage server: Brief case study 10
   1.3 RAID Array Technology: A Useful Insight 11
   1.4 Related Network Technologies 13
   1.5 Scalable Video 20
   1.6 Are SANs an Efficient Solution? 22
      1.6.1 SANs: A brief discussion on salient features 24
      1.6.2 SANs& MSR Technology: a handshake? 26
   1.7 Concluding Remarks 28

2. MULTIPLE SERVERS RETRIEVAL STRATEGY: DATA PARTITIONING APPROACH 29
   2.1 Introduction to Multiple Server Technology 29
      2.1.1 Network architecture 31
      2.1.2 Distinct advantages in using Multiple Server Retrieval 32
2.2 Problem Definition and Preliminary Remarks
   2.2.1 Motivating example
   2.2.2 Some definitions
2.3 Single Installment Retrieval Policy
   2.3.1 Homogeneous channels
   2.3.2 Effect of sequencing on the access time
2.4 Multi-installment Servicing Policy
   2.4.1 Recursive equations and solution methodology
   2.4.2 Homogeneous channels
2.5 Discussions on MSR Strategy
2.6 Concluding Remarks

3. SUPPORTING MULTIPLE CLIENTS: CHANNEL PARTITIONING APPROACH
   3.1 Tuning MSR Scheme for Multiple Clients: Issues and Challenges
   3.2 System Model
      3.2.1 Retrieval process
      3.2.2 About the TGS server
      3.2.3 Some terminology and performance metric
   3.3 Design of the TGS Server
      3.3.1 Task generation strategy
      3.3.2 Constraints imposed by the allowed start times of retrieval
      3.3.3 Movie retrieval scheduling strategy
   3.4 Performance Study
      3.4.1 Simulation experiments
      3.4.2 Performance comparison
   3.5 Concluding Remarks

4. AN ALTERNATIVE RETRIEVAL STRATEGY: A PLAY-WHILE-RETRIEVE APPROACH
   4.1 Service Model and Preliminary Remarks
      4.1.1 Description of the strategy
      4.1.2 Some basic definitions
   4.2 Design of Movie Retrieval Strategies
      4.2.1 Single installment strategy
      4.2.2 Multi-installment strategy
   4.3 Buffer Management at Client Sites
      4.3.1 Analysis on buffer occupancy
      4.3.2 Buffer size availability constraints

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>Problem Definition and Preliminary Remarks</td>
<td>33</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Motivating example</td>
<td>34</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Some definitions</td>
<td>36</td>
</tr>
<tr>
<td>2.3</td>
<td>Single Installment Retrieval Policy</td>
<td>37</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Homogeneous channels</td>
<td>39</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Effect of sequencing on the access time</td>
<td>39</td>
</tr>
<tr>
<td>2.4</td>
<td>Multi-installment Servicing Policy</td>
<td>41</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Recursive equations and solution methodology</td>
<td>41</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Homogeneous channels</td>
<td>43</td>
</tr>
<tr>
<td>2.5</td>
<td>Discussions on MSR Strategy</td>
<td>46</td>
</tr>
<tr>
<td>2.6</td>
<td>Concluding Remarks</td>
<td>51</td>
</tr>
<tr>
<td>3.1</td>
<td>Tuning MSR Scheme for Multiple Clients: Issues and Challenges</td>
<td>55</td>
</tr>
<tr>
<td>3.2</td>
<td>System Model</td>
<td>57</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Retrieval process</td>
<td>57</td>
</tr>
<tr>
<td>3.2.2</td>
<td>About the TGS server</td>
<td>57</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Some terminology and performance metric</td>
<td>60</td>
</tr>
<tr>
<td>3.3</td>
<td>Design of the TGS Server</td>
<td>63</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Task generation strategy</td>
<td>64</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Constraints imposed by the allowed start times of retrieval</td>
<td>68</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Movie retrieval scheduling strategy</td>
<td>75</td>
</tr>
<tr>
<td>3.4</td>
<td>Performance Study</td>
<td>82</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Simulation experiments</td>
<td>83</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Performance comparison</td>
<td>85</td>
</tr>
<tr>
<td>3.5</td>
<td>Concluding Remarks</td>
<td>88</td>
</tr>
<tr>
<td>4.1</td>
<td>Service Model and Preliminary Remarks</td>
<td>94</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Description of the strategy</td>
<td>95</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Some basic definitions</td>
<td>98</td>
</tr>
<tr>
<td>4.2</td>
<td>Design of Movie Retrieval Strategies</td>
<td>99</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Single installment strategy</td>
<td>99</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Multi-installment strategy</td>
<td>103</td>
</tr>
<tr>
<td>4.3</td>
<td>Buffer Management at Client Sites</td>
<td>111</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Analysis on buffer occupancy</td>
<td>111</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Buffer size availability constraints</td>
<td>115</td>
</tr>
</tbody>
</table>
## Contents

4.4 Performance Evaluation of PWR: Single and Multi-installment Strategies  
4.4.1 Behavior of access time 119  
4.4.2 Quantifying client buffer requirements 122  
4.4.3 Effect of load balancing 125  
4.5 Concluding Remarks  
4.5.1 Some open-ended issues 128

5. SCHEDULING OVER UNRELIABLE CHANNELS 131  
5.1 Nothing Is Certain But Death And Taxes! 131  
5.2 A Refined Model 131  
5.3 CBR Case 134  
5.3.1 Single-Installment Case 134  
5.3.2 Multi-Installment Case 140  
5.4 VBR Case 143  
5.5 Optimum Server Ordering 145  
5.6 Determining the Optimum Relaxation Parameter $c$ 150  
5.7 Coupling the Relaxation Parameter $c$ and the Number of Installments $M$ 156  
5.8 Simulation Study 157  
5.9 Concluding Remarks 164

6. ADAPTING TO NETWORK VARIABILITY 169  
6.1 Relaxing the constraints is not enough 169  
6.2 Dynamic Schedule Adaptation 169  
6.3 Single Installment Revisited 172  
6.4 Simulation Study 175  
6.5 Discrete Event Simulation Study 182  
6.6 Concluding Remarks 186

7. FAULT-TOLERANCE ANALYSIS FOR MULTIPLE SERVERS MOVIE RETRIEVAL STRATEGY 189  
7.1 What can go Wrong with an MSR strategy? 189  
7.1.1 Definitions, notations, terminology 191  
7.1.2 Some remarks 193  
7.2 Revisiting the Design and Analysis of MSR Strategy with Server Reliability 194  
7.2.1 Effect of sequencing 196  
7.3 Failure-Recovery Analysis 199
7.3.1 Solution using a deterministic approach 199
7.3.2 Solution using a probabilistic approach 202
7.4 Performance Evaluation and Discussions 204
7.5 Concluding Remarks 211

8. AN AGENT DRIVEN PRACTICAL MSR SYSTEM: JINI TECHNOLOGY 215
8.1 Challenges in Realizing a Practical MSR System 215
8.1.1 Implementation challenges 218
8.2 Service Description and Retrieval Strategy 219
8.2.1 Overview of the service architecture 219
8.2.2 Interaction of a client with various components 221
8.3 Design of the JVoD Software Architecture 229
8.3.1 Movie server : design and architecture 229
8.3.2 Agent : design and architecture 234
8.3.3 Client : design and architecture 240
8.4 Implementation Test-bed and Discussions 242
8.5 Concluding Remarks 247

9. FUTURE RESEARCH DIRECTIONS 251

Index 265
List of Figures

1.1 Illustration of the 6 original RAID levels. Levels 2, 3 and 4 differ mainly in the size of the stripe used (bit, byte and sector accordingly). Levels 2, 3 and 4 have dedicated parity (P) or error correcting code devices. Level 5 distributes the parity sectors over all participating devices. 12

1.2 (a) A multiple distributed server system where all movies are copied in all servers. (b) Limiting the number of copies can provide a greater variety of offerings at the cost of reduced availability. 14

1.3 A multicast based system has two advantages over a unicast based one: reduced server load and reduced traffic between routers. The chaining example shown in (c), uses color coding to indicate the pipeline/virtual batch of machines: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$. 14

1.4 Examples of (a) a batching and (b) a patching system. In (b) patch A is used for client $C_i$ and patch B for client $C_{i+2}$. The time threshold $d$ is displayed as a horizontal bar, following each of the request arrivals. (c) shows a harmonic broadcast arrangement where the video stream is split in 3 parts and broadcasted in the same number of channels/streams. Please note that the medium communicated in Fig. (c) is of much smaller duration than the one assumed in (a) and (b). In particular, the duration of the movie is 3 times the duration of part $S_1$. 17

1.5 Block diagram of a SNR scalable coder. 21

1.6 Block diagram of a temporal scalable coder. 21

1.7 Block diagram of a spatial scalable coder. 22
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.8</td>
<td>Block diagram of a hybrid scalable coder, employing a temporal followed by a spatial decomposition.</td>
</tr>
<tr>
<td>1.9</td>
<td>A typical SAN Architecture showing the interconnection medium with storage and networks nodes</td>
</tr>
<tr>
<td>1.10</td>
<td>Diagram showing the integration of MSR technology with SANs on a small scale multimedia service domain</td>
</tr>
<tr>
<td>2.1</td>
<td>Networked Multimedia Servers Servicing Customers</td>
</tr>
<tr>
<td>2.2</td>
<td>Directed Flow Graph representation for the case of 3 servers.</td>
</tr>
<tr>
<td>2.3</td>
<td>Directed Flow Graph representation using single-installment strategy for multimedia document retrieval from N servers</td>
</tr>
<tr>
<td>2.4</td>
<td>Directed Flow Graph representation using Multi-installment strategy for multimedia document retrieval from N servers</td>
</tr>
<tr>
<td>2.5</td>
<td>Access Time vs Number of multimedia servers using MPEG-I and MPEG-II Streams</td>
</tr>
<tr>
<td>2.6</td>
<td>(a) Number of installments where the access time falls below 1 sec, or where an additional installment would improve the access time less than 5%, against the number of servers and connection speed. (b) The corresponding access times for the installments shown in (a).</td>
</tr>
<tr>
<td>2.7</td>
<td>(a) Number of installments where the access time falls below 1 sec, or where an additional installment would improve the access time less than 5%, against the number of servers and connection speed. The difference from Figure 2.6(a) is that the servers share the same connection to the document requesting party. (b) The corresponding access times for the installments shown in (a).</td>
</tr>
<tr>
<td>3.1</td>
<td>A motivating example for showing the retrieval process of a MSR strategy. The process of retrieving the movie A can be described as follows: (1) The client sends the request for movie A to a neighboring server, e.g., Server 1. (2) Server 1 forwards the request to the Task Generation and Scheduling (TGS) server. (3) TGS server carries out task generation and retrieval scheduling. Subsequently, it multicasts the schedule (i.e., scheduling result) to Server 1 and Server 2. (4) Server 1 and Server 2 deliver the different parts of movie A to the client according to schedule, respectively.</td>
</tr>
<tr>
<td>3.2</td>
<td>System components in the TGS server</td>
</tr>
<tr>
<td>3.3</td>
<td>Flow chart of retrieval process in TGS server</td>
</tr>
<tr>
<td>3.4</td>
<td>Calculation of buffer space requirement</td>
</tr>
<tr>
<td>3.5</td>
<td>Calculation of a movie's access time.</td>
</tr>
</tbody>
</table>
List of Figures

3.6 Aggregate retrieval bandwidth and concurrent retrieval. Each of \( n \) servers renders a part of the portion \( \xi \) of the movie in \( m \) installments. The order in which these retrieved portions are presented is: block 11, block 21, ..., block n1, block 12, block 22, ..., block n2, ..., block 1m, block 2m, ..., block nm. \( N \) sub-channels can be equivalently treated as channel \( \xi \) at every time instant \( t = ixT, i = 1, \ldots, m \).

3.7 Timing diagram of the first playback portion in every channel

3.8 Timing diagram of adjacent playback portions (when \( R_i \leq BW_j \) and \( R_i \leq BW_{j+1} \))

3.9 Timing diagram of adjacent playback portions (when \( R_i \geq BW_j \) and \( R_i \geq BW_{j+1} \))

3.10 Timing diagram of adjacent playback portions (when \( R_i \geq BW_j \) and \( R_i \leq BW_{j+1} \))

3.11 Timing diagram of adjacent playback portions (when \( R_i \leq BW_j \) and \( R_i \leq BW_{j+1} \))

3.12 The timing diagram for Example 1

3.13 Performance comparison of SSRS and MSR strategies under a variation of expected access rates with respect to the actual access rates

3.14 Performance comparison of SSRS and MSR strategies for different allowed maximum access time

4.1 Architecture of a multi-server VoD system

4.2 Example of PWR strategy

4.3 Critical size

4.4 Directed flow graph representation using single installment strategy for movie retrieval from \( N \) servers

4.5 Directed flow graph representation using multi-installment strategy for movie retrieval from \( N \) servers

4.6 Access time vs. number of servers using PWR and PAR: single installment strategy

4.7 Access time vs. number of servers with \( n = 2 \) using PWR and PAR: multi-installment strategy

4.8 Access time vs. number of installments using PWR and PAR: multi-installment strategy

4.9 Client buffer occupancy using PWR and PAR: single installment strategy

4.10 Client buffer size vs. \( \alpha \) using PWR and PAR: single installment strategy
4.11 Client buffer occupancy using PWR and PAR: multi-installment strategy 123
4.12 Client buffer occupancy vs. connection bandwidth using PWR and PAR: multi-installment strategy 124
4.13 Client buffer occupancy vs. number of installments using PWR multi-installment strategy 124
4.14 Loads on servers vs. connection bandwidth using PWR single installment strategy 126
4.15 Loads on servers using PWR multi-installment strategy 126
4.16 Loads on servers vs. connection bandwidth using PWR multi-installment strategy 127

5.1 Example of delivering a document to a client \( C \) by 3 servers, using (a) a single installment strategy, (b) two-installments with a single overhead per server and (c) two-installments with multiple overheads. Figure (a) features a breakdown of the time costs involved, including the extra time provided for transmission (e.g. \((c - 1)bw_0m_0L\) for \( S_0 \)). Colour indexing associates the downloaded parts with portions of the playback operation. 133

5.2 Frame sizes over time for the 78sec MPEG-4 coded trailer of the movie *Matrix Revolutions*. The 640x256 movie was coded with the Xvid codec (http://www.xvid.org). The size of the PCM-coded audio is incorporated in the frame sizes. 144

5.3 The two possible configurations that two servers can be arranged to deliver a document. Inverted patterns depict extra time provided for recovering from network errors. 146

5.4 A plot showing how the individual parts \( m_i \) depend on the value of \( c \). Also shown is a curve for \( c \) as computed from Eq. 5.58. The arrow points to the crossing which represents the optimum value for \( c \) in this particular setup. 154

5.5 Flowchart describing an iterative approach for determining the optimum parameter \( c \). 155

5.6 (a) \( AT \) offered by two identical servers, with \( bw \) ranging from 10 to 20 sec/MB and \( a = 1 \) sec, for a 1 GB movie with \( R = 8 \) sec/MB to a client with an available bandwidth of \( bw_C = 5 \) sec/MB. The number of installments is one of \( M \in \{1, 2, 4, 8, 16\} \). (b) The optimum value of \( c \) is determined via the algorithm in Fig. 5.5, by assuming that \( sz = 1500 \), the confidence interval is 95% and the packet loss probability is 10%. 157
5.7 Algorithm for estimating the optimum $c$ and $M$ parameters given the confidence level $CP$ that the delivery schedule should provide. $conf$ denotes the confidence provided by $c = c_m$.

5.8 Simulation results for a pair of servers, one having one tenth the speed of the other one. Graph (a) shows the access-times for using the pair or just the faster of the two. (b) displays the improvement achieved by utilizing both servers VS only the fast one. Graphs (c) and (d) show the corresponding $c$ and $M$ values as computed from the algorithm in Fig.5.7. The horizontal lines in (c) correspond to the single fast server scenario.

5.9 $AT$ "speedup" for different packet loss probabilities $p$ and number of servers $N$. The $x$ axes display the individual server $bw$, which means that the consumed client bandwidth is $\frac{bw}{N}$.

5.10 $AT$ versus consumed client bandwidth for different packet loss probabilities $p$ and number of servers.

5.11 Bandwidth that has to be devoted by each of a number of identical servers for achieving an access time of 1 minute. The surfaces are pseudo-colored to reflect the corresponding number of required installments.

5.12 $AT$ versus normalized buffer requirements for varying $N$ and packet loss probability. The case of $L = 33MB$ (roughly 5min for $R = 9.31s/MB$) is indicative of movie clips. $M$ is fixed at 1.

6.1 A pull-based scheme that a client can use to adjust the delivery schedule to system changes.

6.2 Algorithm for determining the $AT$ and parts $m_{k,j}$ for $M$ successive deliveries of $L/M$ movie parts (a single overhead is assumed).

6.3 A 3-installment schedule for delivering the "Matrix Revolutions" trailer via 3 identical servers $S(10sec/MB, 0sec)$. It is assumed that $p = 10\%$ and $sz = 1500$ bytes. The corresponding $c$ value is shown above each installment. Also displayed are the $m_{i,j}$ values.
6.4 SIMP optimum access times and number of installments $M$ for delivering a 645MB movie using different portions of the available client bandwidth and a different number of servers sharing that resource. Plots depict different combinations of packet size $sz$ and loss probability $p$. In order to make the graph easier to interpret, we use the normal and not the inverse of the client’s consumed bandwidth in the x-axis.

6.5 SIMP optimum access times and number of installments $M$ for delivering a 33MB movie using different portions of the available client bandwidth and a different number of servers sharing that resource. Plots depict different combinations of packet size $sz$ and loss probability $p$.

6.6 SIMP maximum buffer requirements versus $AT$ for different number of servers and movie size $L$. The buffer requirements are normalized i.e. expressed as a percentage of $L$.

6.7 SIMP VS MISP. The plots display differences between SIMP and MISP corresponding figures, e.g. $AT_{SIMP} - AT_{MISP}$.

6.8 $AT$ difference from an optimum SIMP strategy and associated $M$, when the number of installments is calculated as the threshold that makes any subsequent $AT$ improvements over current estimates, fall below 0.1%. For both document sizes, $sz = 1500B$.

6.9 Speedup over a single server for $N = 2, \ldots, 5$ servers ($L = 645MB$ and $sz = 1500B$).

6.10 An instance of the OMNeT++ based discrete event simulator showing a 4 server setup.

6.11 An illustration of the packet loss probability model of Arai et al. [6] for $min_k = 0$ and $max_k = 0.2$. The vertical lines indicate the different phases $\varphi$ that are used in the discrete event simulation.

6.12 Percentage of cases where playback was not interrupted for (a) $M = 10$, (b) $M = 100$ and (c) $M = 1000$ for different starting phases $\varphi$.

6.13 $AT_{post} - AT$ difference expressed in seconds for different starting phases $\varphi$ and for (a) $M = 10$, (b) $M = 100$ and (c) $M = 1000$.

7.1 Single server($S_j$) failure at time $t_c$. The shadowed area represents the missing data portion $m'_{j}$ that $S_j$ fails to deliver.
List of Figures

7.2 Data portions to be retrieved from the 10 servers for the refined MSR strategy and ideal MSR strategy (denoted as Strategy A in the figure) 205
7.3 Playback rate variation at the client site in case of single server failure during movie retrieval 208
7.4 Access time vs. number of MM servers for the refined MSR strategy and ideal MSR (denoted as Strategy A in the figure) 209
7.5 Comparison of a random sequence and an optimal sequence of servers in terms of data size. The mean availability of all the resources is 90%. 210
7.6 Comparison of a random sequence and an optimal sequence of servers in terms of data size. The mean availability of all the resources is 50%. 211
7.7 Comparison of a random sequence and an optimal sequence of servers in terms of data size. The 18th server in this random sequence is shifted to the 14th position in the optimal sequence. The mean availability of all the resources is 10%. 212
8.1 Overall view of the JVoD software architecture 220
8.2 Flowchart of the ESP strategy 228
8.3 Typical output of the ESP strategy 228
8.4 Movie Server structure design showing various modules 230
8.5 Discovery and Join Message Sequence Chart 231
8.6 Agent structure design showing various modules 236
8.7 Movie Server failure event: broadcast mechanism 236
8.8 Client structure design showing various modules 241
8.9 Accessing JVoD service through the client application - screen shot of the client 241
8.10 Choosing and previewing the movie trailer - screen shot of the client 244
8.11 Retrieving the movie portions - screen shot of the Movie Server 244
8.12 Presentation at the client site - screen shot of the client 245
8.13 Retrieving the movie portions - screen shot of the backup Movie Server 245
8.14 Access time with respect to movie size 246
List of Tables

1.1 List of notations used in continuity constraints derivation 5
3.1 Some common parameters in the VoD system 61
3.2 Channel Partition Algorithm 66
3.3 Known parameters (before calculation) in Example 1 77
3.4 Optimal sizes of the portions for Example 1 78
3.5 Detailed results for Example 1 78
3.6 Parameters in the VoD system simulation 83
3.7 Commonly used equations in VoD system simulation 84
4.1 Coefficients table for multi-installment strategy 107
5.1 Notations 135
Preface

Deriving entertainment and availing a variety of multimedia services via home-computers/laptops or mobile devices, have become commonplace for Internet users in this modern era. Growing high-speed networking technology coupled with the development of ultra-speed multimedia high-end machines facilitates the notion of rendering such media services at attractive costs. Network based multimedia services attempt to render best effort services at cheaper prices. For instance, a video rental store allows users to rent video cassettes, CDs/DVDs at a fixed price. In contrast, a networked multimedia service (NMS) allows a user to surf through a range of collections and obtain the desired content, without having to satisfy timing, or physical location restrictions. Furthermore, a user need not even be confined to a specific location, but could be roaming with a mobile device. Modern Video/Movie-on-Demand (V/MoD) services even allow complete interactivity by supporting functionality that includes variable-speed playback, fast-forward/forward, etc.

In the published literature there are several papers that present comprehensive studies of such NMS systems that are based on a wide spectrum of performance goals and topics. There are several text-books that expose fundamental multimedia technology at undergraduate to graduate level and even serve as guides to practitioners. However, the emphasis in this book is purely on the research perspective of a focused topic in the domain of multimedia. The emphasis in this book is in exposing a specific state-of-the-art NMS technology that is of recent vintage. The service infrastructure we are concerned in this book is for VOD and/or MoD. Although such services have been in place for some time with a varying degree of success, the manner in which these are deployed leaves many things to be desired in terms of supporting large client populations with adaptability, quality of service and low cost. With increasing user demands for multimedia on-demand services, rendering cost effective and reliable services becomes an imperative requirement.
The design of high-fidelity network-based VOD/MoD service infrastructure must carefully consider issues in optimizing various parameters, ranging from the data storage level to customer satisfaction, in terms of providing high quality, reliability and interactivity. To provide an idea of the challenge ahead, since 1997 the Internet traffic has doubled every 6 months, while in the past 5 years hard disk storage has doubled every year. In particular, the number of large size (typically >3GB) multimedia documents keeps increasing on the Internet and experts conjecture that by the year 2005/2006, more than 50% of the information available over the Internet will constitute of large multimedia documents. It is also conjectured that the percentage of requests to such large-volume multimedia document increases in a greater than linear fashion with time. NMSs are attractive from an economic perspective. For instance, in the case of VOD services, depending on the popularity of the movie, the cost per user can be reduced when clever placement of movies on the network is carried out. Thus, the design of a VOD system employs several technologies, ranging from disk arrays, to clever scheduling policies that maximize the use of networked resources.

Contemporary technology, more-or-less uses sophisticated high-end machines to render such NMS. These are typically maintained by the service providers, taking care of services at various physical network domains. Thus, users in a particular domain can avail service when subscribed to a local service provider. However, when a requested movie or a digital media document is not available with a local service provider, it may be fetched from other domains, when such a contract exists between parties when they are from different service providers. In contrast, a recent technology that is introduced in this book, exploits a distributed approach in rendering NMS to the clients by making use of time-and-bandwidth multiplexed strategies and data partitioning strategies. These strategies are shown to be very effective in minimizing large annoying waiting times for the users and simultaneously maximizing the number of customers that a service provider can attract. This technology, during its incipient stages, has stemmed from a mere theoretical interest and has subsequently been conceived as a practically viable scheme in addressing several issues that are akin to NMS elegantly. The key idea is to employ more than one server in rendering NMS to the clients. This technology can be referred to as a Multiple Server Retrieval (MSR) strategy, in general. A variety of issues that are germane to this domain, such as the minimization of access times/waiting times, buffer management, networking issues, fault-tolerance, etc are studied and thoroughly presented. A prototype that has been realized as a (working) proof-of-concept is also presented.

In this book, we expose most of the major research issues and challenges in this MSR media retrieval technology. As a word of caution, this technology
specifically suits retrieving long duration media documents, such as complete movies and Gbytes of both time continuous and discrete media data, over delay sensitive and unreliable network channels. Thus, this technology constitutes an alternative solution to existing problems, opening in the process avenues on how to make use of current Internet-like networks to render such NMSs.

The organization, presentation, and contents of the book are pitched at a research monograph level. The contents are carefully sorted in tune with the specific focus of the book and they are being derived from a number of published papers and articles from the recent literature. As this technology is still in its infancy, the available material is pooled from research efforts of the past 4 to 5 years and peripheral related material from the past 12 years. The mathematical background that is expected to go through the material is modest. Fundamental knowledge of Linear Algebra, Calculus, and Probability theory is expected as these are used throughout the contents, in a variety of occasions, to make the treatment more rigorous and complete. Proofs of all the theoretical claims in the form of Lemmas and Theorems are provided in an emphatic, comprehensive, and step-by-step fashion to render clarity. Several numerical examples in addition to rigorous simulation studies and implementations were provided to clarify all the results. Each chapter has been carefully written to have a continuous flow of contents in a systematic fashion and carries design recommendations for implementation specialists and system level designers. In every chapter, we provide a summary of source material in the form of bibliographic notes.

When it comes to the question of who are the potential users of this monograph, we see a wide spectrum of audience with diversified interests in this multimedia domain. This monograph can be used as a reference text for an advanced undergraduate level course in Multimedia Networking courses and in courses that use NMS topics. For graduate study, this book serves as a directly useful reference in introducing the state-of-the-art technology for students wishing to pursue research in this area. Certain chapters (specifically Chapters 3, 5, 6, and 7) can be a part of advanced graduate level courses. For teachers, additional material and notes are provided in bibliographic notes section at the end of each chapter for quick reference to other allied materials in this domain. This monograph can be used by researchers working directly in this and other related domains such as multimedia networking (at the applications and the network layers), multimedia computing & scheduling, distributed system design, digital document storage and retrieval, to quote a few. Researchers working in the area of Storage Area Networks (SANs) may find this material useful in terms of deriving ideas and algorithms for an alternate implementation, while handling large scale data storage and retrieval. Research organizations and corporate sectors can use this technology to enhance their existing solutions with the use
of an appropriately designed MSR wherever required, as distributed infrastructures and their mastery is what this monograph is all about. For entrepreneurs, certainly this monograph is challenging and entertaining as it exposes several new ideas that can be tuned to fit the current day market and technology trends. For instance, service providers can immensely benefit from this technology in terms of maximizing the number of customers who can subscribe to their NMS, as it has the potential to be a cost-effective and welcomed scheme for customers. Clearly, this technology renders a win-win situation!

In Chapter 1, we first explore a number of different technologies that are related to multimedia retrieval in general. We examine their characteristics from the viewpoint of how they relate to MSR, how they could benefit from MSR or vice-versa, or how they could be superceded by it.

In Chapter 2, we present an introduction to the MSR technology, and the underlying problem setting. We proceed to the design and analysis of a single and multiple installment servicing policy to minimize the access time. This chapter serves as a first, gentle exposure to the ideas underlying MSR design.

In Chapter 3, we extend the study of Chapter 2, to handle multiple clients and discuss a channel partitioning approach. We present a rigorous analytical study to quantify the performance gains which are validated by extensive realistic simulation models.

While Chapters 2 & 3 analyze in detail the design and performance of MSR technology, in Chapter 4, we introduce a modification to the overall approach taken by MSR, that reflects on how video playback can be performed in real-life. Here the client is allowed to initiate the playback soon after a critical portion is downloaded, as opposed to awaiting the completion of the download of an entire portion before kick-starting the playback.

In Chapter 5, we address the issue of packet loss and generic network unreliability that is one of the main issues hampering the deployment of VoD services. We show that small modifications to our mathematical framework suffice to make it capable of producing a robust schedule that is impervious to certain network problems. The trick is to allow the relaxation of the constraints it is based on.

In Chapter 6, we investigate ways for adapting to network variability, in the process extending the robustness of the schedules we can produce. In particular, we compare two competing approaches, one based on a multi-installment strategy and one based on the repetitive application of a single installment strategy. Rigorous discrete event simulations show that the latter can truly offer a robust, adaptable approach to handling network variability.
In Chapter 7 we present fault-tolerance analysis studies pertaining to server crashes and show how availability can be maximized. We specifically present analysis on how to retrieve media data that is lost either due to server crashes or channel failures. Here, we present deterministic and probabilistic approaches in deriving some significant analytical results. The results of this chapter are particularly useful in choosing a set of servers from a given pool depending on reliability/availability criteria.

The results of chapters 5, 6 and 7 indeed demonstrate the fact that MSR is a viable and useful technology to adopt for deploying multimedia services over public networks.

In Chapter 8, we present the design and implementation of a working MSR system based on the Jini platform. We provide a detailed account of the design and implementation of all system components, including the Client, the Server and other essential parts of the service infrastructure.

In Chapter 9, we present our view on the future and scope of MSR and discuss some other dimensions associated with the realization of MSR technologies.
Acknowledgments

We wish to extend our heartfelt thanks to those who have directly and indirectly helped in this book project. We thank the editorial efforts rendered by the Kluwer Academic team and special thanks go to Mr. John Martindale and Ms. Robert Saley in bringing out this book in good shape. We acknowledge all timely transactions over countless emails with John and Saley over the last one year time frame. We would also like to thank Ms. Susan Largerstrom-Fife and Ms. Sharon Palleschi in rendering continuous support from the manuscript preparation stage till its production. We would like to extend our appreciation to Dr. Wujuan Lin of Data Storage Institute of Singapore (DSI), Singapore, for his suggestions on formatting, proof-reading of certain chapters, and suggestions on Chapter 1 contents on SANs. Parts of the work presented in this book have been reviewed earlier by Professor Viktor K. Prasanna, Department of Electrical & Systems Engineering, USC, LA, USA. We would like to thank Professor Prasanna for his effort. Bharadwaj Veeravalli would like to thank Mrs. Xiaorong Li, Mr. Jingxi Jia, and Qiao Yunhai, for their help in formatting and fixing certain figures in Chapters 2, 4, 7, and 8.

On the personal front, Bharadwaj Veeravalli would like to thank his wonderful wife, Lavanya for her patience, understanding, and encouragement rendered throughout the book writing process. He would like to thank his daughters Tanya & Harshitha. He would like to render special thanks to Harshitha for keeping him awake during long night hours, in a way, helping to complete this paramount task on time!! He would like to thank his brother Sivakumar and his family for all the encouragement provided during this time. He would like to render special thanks to his parents-in-law for their unflagging support during this period (2004/2005) without which this task would not have been undertaken. Their help in taking care of the new born and assuming most of the house-hold work helped to squeeze in long writing hours!! He would like to render special thanks to (his Gurus!) Professor Debasish Ghose and Professor
Venkatraman Mani, Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India, who had always a word of encouragement on any research endeavor ventured by him. He would also like to pass a special thanks to his good friend and colleague Professor Tom Robertazzi, Department of EE, SUNY, Stony Brook, NY, USA, who always had a great deal of interest in all his works. Finally, he would like to say a BIG THANK YOU to all his friends in NUS with the Department of ECE, NUS, for providing a conducive and joyful ambience. Special thanks to The Tea Club members who had discussions, almost everyday, on all topics ranging from International affairs to everyday life in Singapore! Thanks Guys!

On a personal note, Gerassimos Barlas would like to thank his parents Dimitrios and Maria for their sacrifices, love and support that helped him go so far. Special thanks go to his dearest wife Katerina for her love, patience, dedication and sacrifice.
Chapter 1

DISTRIBUTED TECHNOLOGIES FOR MULTIMEDIA RETRIEVAL OVER NETWORKS

1.1. Video on Demand: The Challenge and Contemporary Solutions

Video or Movie on Demand (VoD/MoD) has been touted for a long time as the next best thing to be delivered over the public Internet. The problem is that the size of the data involved and the strict timing constraints that must be maintained are overwhelming challenges that have limited VoD to proprietary networks or niche markets.

The quality of offered services in the VoD domain is usually measured by:

- Access Time: how much time the client has to wait between making the request and the beginning of the actual playback (also referred to as initiation latency [39])

- Movie Quality: the number or interruptions, artifacts or distortions present in the playback due to network errors, packet losses, etc.

Although there is no universally acceptable quality (or distortion) metric [111], it is natural for a client to expect the highest possible quality, or at least the quality promised by the Service Provider.

If one considers only the elimination of playback artifacts, storing the movie before actual playback is an option. However, the delay involved and associated problems in Digital Rights Management (DRM) make this option a very unpopular one. Hence, the persistence of the research community in perfecting streaming in its various shapes and forms. Streaming, which is the popular term used to refer to concurrent playback and download, is not without problems as anyone who has actually tried it will attest.
A few examples can shed light to the problems involved:

- An MPEG-2 coded, DVD quality, feature-length movie has a typical bit rate of 7.5Mbps. If we assume that the movie is 100 minutes long and the playback must commence at the most 5 minutes after the client requests the particular media, then the above characteristics and requirements translate to the need to communicate a 5.24 GB document at a rate of 872 Kbytes/sec!

- Although the above example is extreme, moving to an advanced codec (e.g. MPEG-4) coupled with a modest SIF resolution (352x240 pixels), would enable the use of bit rates in the range of 300 kbps. If we again assume that the movie is 100 minutes long and the playback must commence 5 minutes after the client makes the request, then the client machine needs to receive a total of 214.58 MB at a rate of 34.88 Kbytes/sec which is still not a trivial task while at the same time the quality offered is a far cry even from TV-broadcast quality.

Past experience has shown that just “buffering" the data, i.e. storing a portion of the data before playback can start, is only part of the solution in VoD service deployment. The unpredictability of the communication media and the bulk of the data involved, require either special conditioning by the underlying network (e.g. multicasting) or dedicated architectures to be deployed (e.g. simulcasting). Contemporary solutions to these problems try to avoid the creation of bottlenecks or hot-spots in the network, by employing multiple “entities" in various stages of the data delivery process. The manifestations of this approach are:

- Scalable video : a video stream can be split into multiple streams, each carrying a piece of the information needed to reconstitute the original. Typically, a base and detail streams are created, where the detail stream can provide better temporal and/or spatial resolution but it is not required for decoding the base stream.

- Multicasting : in the case of a real-time video feed that is of interest to a wide audience, multicasting can provide the “goods" to multiple recipients with a minimum of network overhead.

- Simulcasting : was conceived as a unicast-based alternative to multicasting that uses the clients as repeaters and requires no special network support in the fashion that multicasting does. Several schemes have been proposed for building and maintaining the tree of participating nodes. It is also known as overlay or peer-to-peer (P2P) multicast.

- Multiple distributed servers: the content is delivered in disjoint parts by multiple servers, thus allowing for dynamic adaptation to network conditions and server loads.
In tandem with the above methods and depending on the possibility of data loss and its impact on the provided service quality, error correction and error concealment techniques can be utilized. In the sections that follow, each of the above techniques is presented and discussed in greater length.

The focus of this book is on the multi-server approach and what it has to offer towards realizing VoD over the public Internet. The distributed multi-server approach provides scalability and fault tolerance at the network connection level, as parallel video servers offer both at the storage level [66].

1.2. General Multimedia Storage and Retrieval Architectures

Under a network based service infrastructure, we will now study the following:

- Issues and challenges in the design of multimedia storage servers\(^1\)
- Different types of media delivery architectures and certain important criteria for jitter free presentation
- Basics principles in the design of admission control algorithms

Specifically, we will study some possible storage server architectures, the nature of data/objects being stored, and derive admissibility criteria for a jitter-free high quality service.

1.2.1 Multimedia servers: A resource management perspective

Future advances in multimedia technology will make it feasible for distributed systems to support a range of multimedia services on networks. Already there are a host of applications running on the network, however, somewhat restricted in their service abilities. Most of the systems are dedicated to cater a single service to a subscribed community. However, technology still needs to be improved in order to make a single server support a range of applications. As a networked multimedia server is expected to serve a large pool of clients, it is possible to view this server as a resource. In the following, we shall see how a single multimedia server manages the storage of media documents and present an admission control procedure that cleverly adapts to a large client pool.

1.2.1.1 Storage requirements

Multiple data streams: A multimedia object may consist of three types of components: audio, video, and text. By and large, these three components

---

\(^1\)These are referred to as VoD/MoD servers or just plain multimedia servers in the literature.
are separate when captured or composed and they could be handled as three
different streams. Of course, conventional movies are exceptional to this case.
Similarly, during the retrieval and delivery, these three streams are routed to
three different output devices. Storing these media may entail additional pro­
cessing for combining them during storage (multiplexing), and for separating
them during retrieval at the time of delivery (demultiplexing). On the other
hand, if the three media are stored separately, what needs to be stored is the
temporal relationships between the media types so as to ensure proper synchro­
nization between them during retrieval.

Continuous recording and retrieval of data streams: Recording and play­
back of motion video and audio are time continuous operations. The file system
must organize the multimedia data on the disks so as to guarantee that their stor­
age and retrieval proceed at their respective real-time rates.

Large size files: In general, the audio and video require very large storage
spaces. If the file system is to act as a basis for supporting media services such
as document editing, mail, distribution of news, VoD, entertainment, etc., it
must provide mechanisms for manipulating and sharing stored data. For these
mechanisms to be efficient on large data, disk access must be stream-lined. The
design of a file system that addresses the above issues is what a multimedia stor­
age server is all about. We will first see some most commonly used terminology
and notations used in this domain.

1.2.1.2 Some most commonly used terminology and notations
Following is a list of commonly encountered terms in video coding:

- **Frame**: A basic unit of video.
- **Sample**: A basic unit of audio.
- **Strand**: is an immutable, sequence of continuously recorded audio samples
  or video frames. The immutability of strands is a necessary condition to
  simplify the process of garbage collection.
- **Block**: is the basic unit of disk storage. There are two types of blocks: (i)
  *Homogeneous* and (ii) *Heterogeneous*. In the case of (i), all the data belong
to only one type of media and in (ii), the data contains multiple media.
- **Rope**: A collection of multiple strands (of same and different media) tied
together by synchronization information.

Table 1.1 summarizes a list of notations that will be used in subsequent para­
graphs for deriving continuity constraints. Thus, with the above notations, we
can write the expression for
Table 1.1. List of notations used in continuity constraints derivation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Explanation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{ar}$</td>
<td>Audio recording rate</td>
<td>samples/sec</td>
</tr>
<tr>
<td>$R_{vr}$</td>
<td>Video recording rate</td>
<td>frames/sec</td>
</tr>
<tr>
<td>$R_{dr}$</td>
<td>Rate of data transfer from the disk</td>
<td>bits/sec</td>
</tr>
<tr>
<td>$R_{vd}$</td>
<td>rate of video display</td>
<td>bits/sec</td>
</tr>
<tr>
<td>$\eta_{vs}$</td>
<td>Granularity of video storage</td>
<td>frames/block</td>
</tr>
<tr>
<td>$\eta_{as}$</td>
<td>Granularity of audio storage</td>
<td>samples/block</td>
</tr>
<tr>
<td>$S_{vf}$</td>
<td>Size of the video frame</td>
<td>bits/frame</td>
</tr>
<tr>
<td>$S_{as}$</td>
<td>Size of audio sample</td>
<td>bits/sample</td>
</tr>
<tr>
<td>$l_{ds}$</td>
<td>Scattering parameter</td>
<td>sec</td>
</tr>
</tbody>
</table>

(i) The duration of the playback of a video block as, $\eta_{vs}/R_{vr}$.

(ii) The total delay to read a video block from a disk as, $l_{ds} + (\eta_{vs}S_{vf})/R_{dr}$

(iii) The time to display a video block as, $(\eta_{vs}S_{vf})/R_{vd}$. Note that this time is for decompression and Digital-to-Analog conversion.

### 1.2.1.3 Continuity requirements

For continuous retrieval of media data, it is essential that the media information are available at the display device at or before the time of playback. We refer to this as the continuity requirement or continuity constraint. If this constraint is violated, then the displaying device will starve for the data and the presentation continuity will be lost. We now analyze three kinds of service architectures - sequential, pipelined, and concurrent architectures for continuity requirements.

**Sequential Architectures:** These architectures serialize the read and display (similarly capture and store) operations. Each block is transferred from the disk to a buffer in the video device, and then displayed before initiating the transfer of the next block. The continuity requirement is met in this case if the sum of time to read a block from disk and the time to display it does not exceed the duration of its playback. That is,

$$
\left( l_{ds} + \frac{\eta_{vs}S_{vf}}{R_{dr}} \right) + \frac{\eta_{vs}S_{vf}}{R_{vd}} \leq \frac{\eta_{vs}}{R_{vr}}
$$

(1.1)

**Pipelined Architectures:** These perform read and display operations in parallel. If there are a minimum of two buffers on the video device, one holding the