Preface

This book explains the techniques to store and retrieve multimedia information in multimedia storage systems. It describes the internal architecture of storage systems. Readers will be able to learn the internal architectures of multimedia storage systems. Many techniques are described with details. Examples are provided to help readers understand the techniques. By understanding these techniques, we hope that readers may also apply similar techniques in the problems that they encounter in their everyday life. In particular, this book would be helpful to managers who wish to improve the performance of their multimedia storage systems.

To the best of our knowledge, there are many books about multimedia information and only a few books discuss the storage systems in detail. Only one of them describes the storage and retrieval methods for multimedia information. However, none of them have discussed the storage and retrieval methods in hierarchical storage systems. Therefore, we consider it necessary to explain the storage techniques for multimedia information on storage systems and hierarchical storage systems in a new book. This book discusses the research on multimedia information storage and retrieval techniques.

This book focuses on the storage and retrieval methods. Some other techniques, though somewhat related, are however outside the scope of this book. Those topics include security of multimedia data in the storage systems, protocols to deliver multimedia information across the networks, and real time processing of multimedia information. Readers can easily find these topics from other books.

This book is divided into the following six sections:

- 1. Background information in Section I.
- 2. Data placement on disks in Section IIa.
- 3. Data placement on hierarchical storage systems in Section IIb.
- 4. Disk scheduling methods in Section III.
- 5. Data migration methods in Section IV.
- 6. Cache replacement policies in Section V.

We start this book with the background of multimedia storage technology in Section I. Multimedia applications process digital media that were only present in the entertainment industry. Multimedia information systems process digital media data according to the needs in these applications. Data compression is vital to the success of multimedia information systems and we explain two image and video compression standards. Traditional storage systems need to be enhanced or improved to support the data storage and retrieval operations. The characteristics of multimedia access patterns have significant impacts on the performance of the storage systems.

In Section IIa, "Data Placement on Disks," we describe the data placement methods that organize the storage locations of multimedia data on disks. Data placement methods organize the multimedia data according to the characteristics of multimedia data access patterns. New techniques have been designed to improve the performance of multimedia storage servers to an acceptable level. Data placement methods are grouped according to the strategies being applied, including statistical placement, striping, replication, and constraint allocation.

In Section IIb, "Data Placement on Hierarchical Storage Systems," we describe the storage organization of multimedia data on hierarchical storage systems. Data placement methods have been designed to achieve efficient retrievals of multimedia data. The data placements are categorized according to the strategy in use, including contiguous placement, statistical placement, striping, and constraint allocation.

In Section III, "Disk Scheduling Methods," the disk scheduling methods that rearrange the service sequences of the waiting requests are described. The methods that schedule normal disk requests are first described. The feasibility conditions to merge concurrent streams are then followed. After that, we describe the scheduling methods for streams of multimedia requests.

In Section IV, "Data Migration," we show the methods to migrate data across the storage levels of the hierarchical storage systems. Data residing on the hierarchical storage systems are migrated from high levels with high access latency to lower levels with low access latency. Staging methods move multimedia objects across the storage level via staging buffers. Time slicing method accesses objects in time slices in order to reduce the start-up latency of streams. Pipelining methods minimize the start-up latency and staging buffer size for multimedia streams.

In Section V, "Cache Replacement Policy," the cache replacement methods of multimedia servers are described. Efficient cache replacement policies on these servers keep the objects with high access probability on the cache. They improve the cache replacement methods of multimedia streams so that multimedia data can be delivered efficiently over the Internet. Memory caching methods replace objects with low cache value so that high cache value objects can be kept for efficient cache performance. Stream dependent caching methods assign cache values to object segments in order to improve the cache efficiency for multimedia objects. Cooperative proxy servers share their Web cache contents so that the cache performs efficiently when similar objects are accessed by their clients.

The organization of chapters in this book is as follows:

- 1. Background in Section I.
 - a. Introduction in Chapter I.
 - b. Multimedia information in Chapter II.
 - c. Architectures of storage systems in Chapter III.
 - d. Data compression techniques and standards in Chapter IV.
- 2. Data placement on disks in Section IIa.
 - a. Statistical placement on disks in Chapter V.
 - b. Striping on disks in Chapter VI.
 - c. Replication placement on disks in Chapter VII.
 - d. Constraint allocation on disks in Chapter VIII.
- 3. Data placement on hierarchical storage systems in Section IIb.

- a. Tertiary storage devices in Chapter IX.
- b. Contiguous placement on hierarchical storage systems in Chapter X.
- c. Statistical placement on hierarchical storage systems in Chapter XI.
- d. Striping on hierarchical storage systems in Chapter XII.
- e. Constraint allocation on hierarchical storage systems in Chapter XIII.
- 4. Disk scheduling methods in Section III.
 - a. Scheduling methods for disk requests in Chapter XIV.
 - b. Feasibility conditions of concurrent streams in Chapter XV.
 - c. Scheduling methods for request streams in Chapter XVI.
- 5. Data migration in Section IV.
 - a. Staging method in Chapter XVII.
 - b. Time slicing method in Chapter XVIII.
 - c. Normal pipelining in Chapter XIX.
 - d. Space efficient pipelining in Chapter XX.
 - e. Segmented pipelining in Chapter XXI.
- 6. Cache replacement policies in Section V.
 - a. Memory caching methods in Chapter XXII.
 - b. Stream dependent caching in Chapter XXIII.
 - c. Cooperative Web caching in Chapter XXIV.

In Chapter I, "Introduction," we give an overview of the techniques that are covered in this book. The techniques are described briefly according to the division of parts in this book.

In Chapter II, "Multimedia Information," we start with describing the characteristics of multimedia data. Some applications that are involved in using and processing multimedia information are listed as examples. The representations of multimedia data show how the large and bulky multimedia data are represented and compressed. The multimedia data are also accessed in request streams. Readers who are familiar with multimedia processing may skip this chapter. In Chapter III, "Storage System Architectures," the architectures of storage systems are explained. Multimedia systems are similar to traditional computers systems in term of their architectures. Multimedia computer systems are built with stringent processing time requirements. The components of the computer system, including the storage servers, need to process a large amount of data in parallel within a guaranteed time frame. The storage server needs to access data continuously to the clients according to the clients' requests. Multimedia objects are large and the magnetic hard disks need to access segments of the objects within a short time. These requirements lead to the emergence of constant recording density disks and zoned disks. Readers who have deep understandings of the computer storage architectures may skip some descriptions and go to the performance equations immediately.

In Chapter IV, "Data Compression Techniques and Standards," the data compression techniques and standards are described. We describe the general compression model, text compression, image compression and JPEG2000, and video compression and MPEG2. These data compression techniques are helpful to understand the multimedia data being stored and retrieved.

In Chapter V, "Statistical Placement on Disks," two statistical placement methods are described. The statistical placement strategy is based on the difference in access characteristics of the multimedia streams. The frequency based placement method optimizes the average request response time. It uses an algorithm to place the objects according to their access frequencies. The bandwidth based placement method places objects according to their data rates. The storage system maintains its optimal performance according to the object data transfer time without reorganizations. Readers may find this chapter useful in other situations which involve probabilities.

In Chapter VI, "Striping on Disks," three striping methods are explained in detail. Multimedia streams need continuous data supply. The aggregate data access requirement of many multimedia streams imposes very high demand on the access bandwidth of the storage servers. The disk striping or data striping methods spread data over multiple disks to provide high aggregate disk throughput. The simple striping methods increase the efficiency of serving concurrent multimedia streams. Multimedia streams access the data stripes according to their actual data consumption rates. The disk bandwidth and the memory buffer are used efficiently. The staggered striping method provides effective support for multiple streams accessing different objects from a group of striped disks, and it automatically balances the workload among disks. The pseudorandom placement method maintains that the data stripes are evenly distributed on disks and it reduces the number of data stripes being moved

when the number of disks increases or decreases. It reduces the workload on data reorganization when disks are added or removed.

In Chapter VII, "Replication Placement on Disks," several replication placement methods on disks are shown. When extra storage space is available, the storage system may keep extra copies of the stored objects. Extra copies of objects may be able to increase the storage system performance. The recent trend of technology shows that storage capacity is increased at a faster pace than the access bandwidth. Storage capacity may not be a problem when compared to the access bandwidth. The replication strategy applies redundancy to increase reliability of the storage system and availability of the stored objects. It reduces network load, start-up latency. It avoids disk multitasking. It maintains the balance of space and workload.

In Chapter VIII, "Constraint Allocation on Disks," two constraint allocation methods are described. Constraint allocation methods limit the available locations to store the data stripes. They reduce the overheads of serving concurrent streams from the same storage device. The maximum overheads in accessing data from the storage devices are lowered. When many streams access the same hot object, the phase based constraint allocation supports more streams with less seek actions. The region based allocation limits the longest seek distance among requests.

In Chapter IX, "Tertiary Storage Devices," the tertiary storage devices are detailed. Several types of storage devices, including magnetic tapes, optical disks, and optical tapes, are available to be used at the tertiary storage level in hierarchical storage systems. These storage devices are composed of fixed storage drives and removable media units. The storage drives are fixed to the computer system. The removable media unit can be removed from the drives so that the storage capacity can be expanded with more media units. When data on a media are accessed, the media unit is accessed from their normal location. One of the storage drives on the computer system is chosen. If there is a media unit in the storage drive, the old media unit is unloaded and ejected. The new media unit is then loaded to the drive. Readers who are familiar with the robotic tape libraries may skip this chapter and directly move on to the placement methods.

In Chapter X, "Contiguous Placement on Hierarchical Storage Systems," two contiguous placement methods are described. The contiguous placement is the most common method to place traditional data files on tertiary storage devices. The storage space in the media units is checked. The data file is stored on a media unit with enough space to store the data file. When tertiary storage devices are used to store multimedia objects, the objects are

stored and retrieved similar to traditional data files. Since the main application of the tertiary storage devices is to back up multimedia objects from computers, the objectives of the contiguous method are (1) to support back up of multimedia objects efficiently and (2) to reduce the number of separate media units that are used to store an object.

In Chapter XI, "Statistical Placement on Hierarchical Storage Systems," we describe the statistical strategy to place multimedia objects on hierarchical storage systems. The objective of the data placement methods is to minimize the time to access object from the hierarchical storage system. The statistical strategy changes the statistical time to access objects so that the mean access time is optimal. The frequency based placement method differentiates objects according to their access frequencies. The objects that are more frequently accessed are placed in the more convenient locations. The objects that are less frequently accessed are placed in the less convenient locations.

In Chapter XII, "Striping on Hierarchical Storage Systems," two striping techniques are explained with details. The data striping technique has been successfully applied on disks to reduce the time to access objects from the disks. Thus, the striping technique has been investigated to reduce the time to access objects from the tape libraries in a similar manner. Similar to the striping on disks, the objective of the parallel striping method is to reduce the time to access objects from the tape libraries. The parallel tape striping directly applies the striping technique to place data stripes on tapes. The triangular placement method changes the order in which data stripes are stored on tapes to further enhance the performance.

In Chapter XIII, "Constraint Allocation on Hierarchical Storage Systems," two approaches to provide constraint allocations on different types of media units are described. Multimedia objects are large in size, but the access latency of hierarchical storage systems is high. The hierarchical storage systems need to provide high throughput in delivering data. Multimedia streams should be displayed with continuity. Depending on the data migration method, the whole object or only partial object is retrieved prior to the beginning of consumption. The constraint allocation methods limit the freedom to place data on media units so that the worst case would never happen. They reduce the longest exchange time and/or the longest reposition time in accessing the objects. The interleaved contiguous placement limits the storage locations of data stripes on optical disks. The concurrent striping method limits the storage locations of data stripes on tapes.

In Chapter XIV, "Scheduling Methods for Disk Requests," two common disk scheduling methods are explained. Disk scheduling changes the sequence

order to serve the requests that are waiting in the queue. While data placement reduces the access time of a disk request, scheduling reduces the waiting time of a request. The longer the waiting queue, the more useful is the scheduling method. When there are not any requests in the waiting queue, any scheduling methods perform the same. A disk scheduling policy changes the service order of waiting requests. It accepts the waiting requests and serves them in the new service sequence. The first-in-first-out policy serves requests in the same order as the incoming order of the waiting requests. The SCAN scheduling method serves the waiting requests in the order of their accessing physical track locations to serve the requests efficiently.

In Chapter XV, "Feasibility Conditions of Concurrent Streams," we prove the feasibility conditions to accept homogeneous and heterogeneous streams to a storage system. Multimedia storage systems store data objects and receive streams of requests from the multimedia server. When a client wishes to display an object, it sends a new object request for the multimedia object to the multimedia server. The multimedia server checks to see if this new stream can be accepted. The server encapsulates the data stripe of the accepted streams as data packets and sends them to the client. The server sends data requests periodically to the storage system. Each of these data requests has a deadline associated with it. Every request of a stream, except the first one, must be served within the deadline to ensure continuity of the stream. We prove that heterogeneous streams can be accepted when their streams accessing patterns satisfy the feasibility conditions. Readers may skip the proofs of the equations in this chapter in the first reading.

In Chapter XVI, "Scheduling Methods for Request Streams," we describe three scheduling methods for multimedia streams of requests. These scheduling methods use either serve requests according to their deadline or serve the stream in round robin cycle in order to provide real-time continuity guarantee. They all use the SCAN scheduling method to improve the efficiency in serving requests. The earliest deadline first scheduling method serves requests according to their deadlines so that the requests would not wait too long and miss their deadlines. The SCAN-EDF scheduling method serves requests with the same deadline in the SCAN order. It improves the efficiency of the storage system using the EDF scheduling method. The group sweeping scheduling method serves groups of streams in round-robin cycles. It improves the efficiency of the storage system and provides realtime continuity guarantees to the streams. It is also fair to all the streams by serving one request of every stream in each cycle. In Chapter XVII, "Staging Methods," we describe one of the data migration methods. Data migration is the process of moving data from tertiary storage devices to secondary storage devices in hierarchical storage systems. The three approaches to migrate multimedia data objects across the storage levels are staging, time slicing, and pipelining. The staging method accesses an object using two stages. The staging method is simple and flexible. It is suitable for any type of data on any tertiary storage systems. Some readers may find the staging method is simple and just browse through this chapter.

In Chapter XVIII, "Time Slicing Method," the time slicing method is described. Tertiary storage devices provide huge storage capacity at low cost. Multimedia objects stored on the tertiary storage devices are accessed with high latency. The time slicing method is designed to reduce the start up latency in accessing multimedia objects from tertiary storage devices. The start-up latency is lowered by reducing the amount of data being migrated before consumption begins. The time slicing method accesses objects at the unit of slices instead of objects. Streams can start to respond at an earlier time.

In Chapter XIX, "Normal Pipelining," the first pipelining method is introduced. Three pipelining methods, including normal pipelining, space efficient pipelining, and segmented pipelining, can be used to access multimedia objects with minimal start-up latency. Apart from reducing the start up latency, the pipelining methods also reduce the usage of the staging buffers. The normal pipelining method finds the minimum fraction of the object before the stream can start to display it. The formula to find minimum size of the first slices is explained. The pipelining method minimizes the start-up latency for the tertiary storage devices whose data transfer rate is lower than the data consumption rate of the objects.

In Chapter XX, "Space Efficient Pipelining," the space efficient pipelining method is explained. The space efficient pipelining method is designed for pipelining objects from low bandwidth storage devices for display. It re-trieves data at a rate lower than the data consumption rate. It keeps the front part of objects resident on disk cache to start a new stream at disk latency. It uses the disk space efficiently to handle more streams. The basic policy reuses the circular buffer to store the later slices of the objects. The shrinking buffer policy reduces the circular buffer size after a slice is displayed. It is particularly useful when the circular disk buffer constraint is tight. The space stealing policy reuses the storage space containing the head of the object as part of the circular buffer.

In Chapter XXI, "Segmented Pipelining," the segmented pipelining method to reduce the latency in serving interactive requests is presented and analyzed.

The segmented pipelining method divides objects into segments and slices so that the object can be pipelined from the hierarchical storage system. The segmented pipelining method is analyzed in terms of disk space requirement and the reposition latency. It uses small extra disk space to support object previews and efficient interactive functions. It can offer extra flexibility in controlling the amount of disk space usage by adjusting the storage location of the preload data. The segmented pipelining is an efficient and flexible data migration method for the multimedia objects on hierarchical storage systems.

Multimedia objects can be stored in the content servers on the Internet. When clients access multimedia objects from a content server, the content server must have sufficient disk and network to deliver the objects to the clients. Otherwise, it rejects the requests from the new clients. The server and network workloads are important concerns in designing multimedia storage systems over the Internet. The Internet caching technique helps to reduce the number of repeated requests for the same objects from popular content servers. As caching consumes myriad storage space, the cache performance is significantly affected by the cache size. Cache admission policies determine whether a newly accessed object should be stored onto the cache devices. Cache replacement policies decide which objects should be removed to release space. The cache replacement policy can be divided into memory caching and stream dependent caching.

In Chapter XXII, "Memory Caching Methods," we describe several replacement policies in memory caching. Memory cache replacement policies assign a cache value to each object in the cache. This cache value decides the priority of keeping the object in the cache. When space is needed to store a new object in cache, the cache replacement function will choose the object with the lowest cache value and delete it to release space. The objects with high cache values will remain in the cache. Different cache replacement policies assign different cache values to the objects. The traditional LRU method keeps the objects that are accessed most recently. It is simple and easy to implement and the time complexity is very low. The LFU, LUV, and mix methods keep track of the object temperature and remove the coldest objects from the cache first. The LRU-min, GD-size, LUV, and mix methods keep the small and recently accessed objects in the cache. The GD-size, LUV, and mix methods also include latency cost of objects in the cache to lower the priority of objects that can be easily replaced.

In Chapter XXIII, "Stream Dependent Caching," the stream dependent caching methods that guarantee continuous delivery for multimedia streams

xxii

are described. The storage techniques on stream dependent caching include resident leader, variable length segmentation, video staging, hotspot caching, and interval caching. They will divide each multimedia object into smaller segments and store selected segments on the cache level. The resident leader method trades off the average response time of requests to reduce the maximum response time of streams. The variable length segmentation method divides the objects into segments of increasing length so that large segments may be deleted to release space more efficiently. The video staging method retrieves high bandwidth segments to reduce the necessary WAN bandwidth for streaming. The hotspot caching method creates the hotspot segments of objects to provide fast object previews from local cache. The interval caching method keeps the shortest intervals of video to maintain the continuity of streams from the local cache content. The layer based caching method adapts the quality of streams to the cache efficiency. It uses the continuity and completeness as metrics to measure the suitability of the caching method for multimedia streams. The cost based method for wireless clients reduces the quality distortion over the error-prone wireless networks with the help of the cache content. The cache values of the segments are composed of the network cost, the start-up latency cost, and the quality distortion cost.

In Chapter XXIV, "Cooperative Web Caching," we describe how Web caches cooperate to raise the overall cache performance on the Internet. Hierarchical Web caching reduces network latency on requests. Front and rear partitioning reduces the start-up latency of streams. Directory based cooperation avoids the contention on parent proxy server. Hash based cooperation achieves low storage overheads and update overheads. Multiple hotspot caching keeps the hotspot blocks to provide fast local previews. The performances of various object partitioning methods in cooperative multimedia proxy servers are analyzed.