Multimedia Data Elevation under a Hierarchical Storage Model

Athena Vakali and Evimaria Terzi

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907, USA

Abstract. Multimedia data storage is a critical issue in large scale applications. This paper proposes a frequency based multimedia data representation model which effectively guides data storage and elevation among the secondary and tertiary storage levels. Multimedia data are stored on the tertiary storage level and (based on certain popularity criteria) they are elevated on secondary level towards improving both the request servicing and the data's accessibility. The proposed multimedia data elevation is a prefetching approach since it is performed "a priori" (not on demand) based on available information on users access patterns. Secondary storage placement is performed by the use of two distinct type placement policies, namely the "Constructive Placement" and the "Iterative Improvement" algorithms. A simulation model has been developed to evaluate the proposed hierarchical data model and the applied placement strategies. Experimentation results have shown that the this hierarchical approach under the iterative improvement placement outperforms earlier related multimedia data placement policies.

Keywords: multimedia data storage, tertiary and secondary storage levels, hierarchical storage subsystems, data placement algorithms.

1 Introduction

Physical storage of multimedia objects is a challenging problem due to the two principal constraints of multimedia data: size and timing. An appropriate model is critical for multimedia data representation due to the variety of types of data and their requirements in terms of space, time and complexity. In [1] a classification of such representation models, based on the notion of time is presented. The main classification involves the *timeline*, the *interval-based* and the *constraint-based* models.

The main issues related to multimedia storage are identified in [10] whereas details on the implementations of multimedia and Video On Demand storage servers are given in [2]. Hierarchical multimedia storage has been proposed due to multimedia upscale space requirements and the storage hierarchies include both the secondary and the tertiary levels. *Secondary storage level* usually involves single or multiple disks configurations and various disk modeling and performance issues have been investigated extensively [7, 13, 16, 17]. Different multimedia data placement schemes on disk systems have been studied in [3]. Secondary storage systems consisting of more than one disks, (disk arrays), have been proved to improve the overall system's performance. *Tertiary* *storage level* involves tapes and multi-tape topologies. A state of the art regarding tertiary storage research is given in [12]. Research works in [4, 5] evaluate and discuss storage hierarchies and the usefulness of current tertiary storage systems for several new types of applications. Furthermore, tertiary storage level data placement has been investigated in several research efforts [5, 15, 22].

Exploitation and interaction between secondary and tertiary storage levels have been proposed and have been proven to be beneficial to the storage system's functionality and responsiveness as shown in [18], where continuous data are elevated from their permanent place in tertiary to the secondary level for better display purposes.

This paper presents a model for effective multimedia data accessing and request servicing. The present work is focusing on studying effective multimedia data placement employed under different storage levels and by involving the user access patterns and preferences in the storage policy. The multimedia data representation model is based on the idea of the models presented by the authors in [20, 21] and the multimedia data are stored in both secondary and tertiary storage levels as proposed in [18]. Since most current multimedia data applications involve browsing and accessing among different multimedia data objects, the frequency of access to data is depicted as a valuable guide for the storage approach.

The remainder of the paper is organized as follows. The next section introduces the proposed multimedia data representation model. The data elevation policies are presented in Section 3, whereas the data placement algorithms performed in Secondary Storage level are discussed in Section 4. Section 5 has experimentation results and conclusions and further research topics are given in Section 6.

2 The Multimedia Data Representation Model

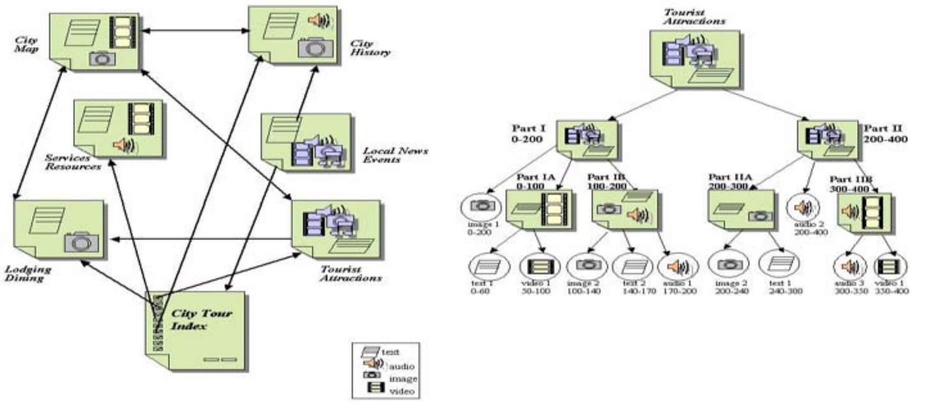

Fig. 1. The navigational multimedia data graph and the multimedia object analysis.

The considered multimedia applications involve interaction and interconnection between various multimedia objects over which users navigate. Left part of Figure 1 presents an indicative navigation example over a multimedia city virtual tour, while the right part presents an analysis of one of the multimedia objects to its (physically stored) components with their associated display times. In the example of Figure 1 *Tourist Attraction* is the multimedia object and objects such as *image 1, text 1, video 1, ...* are physical objects. There are two types of objects in that kind of applications namely *Physical* and *Multimedia Objects* already defined in [20, 21]. The *Physical Objects (PObjs)* that are specific data type entities that correspond to physical storage entities expressed in a number of blocks of the a storage medium. The *Multimedia objects (MObjs)* are sets of various *PObj*s related by their display at specific time intervals in a multimedia data stream.

The browsing graph model presented in [20, 21] is appropriate for the considered navigational multimedia model since it captures the interactive nature of multimedia applications. Since a user navigates from one multimedia object to another the user access pattern can be represented by the arcs weights. These weights correspond to the probabilities of navigating from one node to another. As proven in [3, 20, 21] the original browsing graph can be transformed to an undirected graph and each node can be characterized by a value which corresponds to the frequency of access to that node. The vector of access frequencies $f \;=\; (f_1, \cdots, f_M)$ (where f_i is the frequency of node i), is calculated by the formula:

$$f \;=\; \lim_{K \to \infty} P^K$$

where $P \;=\; [p_{i,j}]$ is the probabilities array of visiting nodes $1 \leq i, j, \leq M$, or the adjacency matrix for the specific weighted undirected graph.

The popularity of *PObjs* is defined as follows:

Definition 1 : The *popularity* of *PObj* x participating in such an application is given by the formula :

$$pop[x] = \sum_{i=1}^{M} f_i \times np_{ix}$$

where f_i is the frequency of access of the ith $MObj$ and np_{ix} is the number of object x playouts in node i. It is obvious that the popularity of a *PObj* is higher when this object is part of a “popular” node and when it should be played for several times within this node.

3 The storage topology and the Data Elevation Approach

Functionality and advantages of adopting a hierarchical storage management system are extensively discussed in [18]. Here we consider a hierarchical storage system of the following configuration :

- *Tertiary Storage* : this level consists of a tertiary storage library. The considered library has one robot arm, which is capable of moving between any tape stored in the library while the tapes are assigned to drives in order to be played.

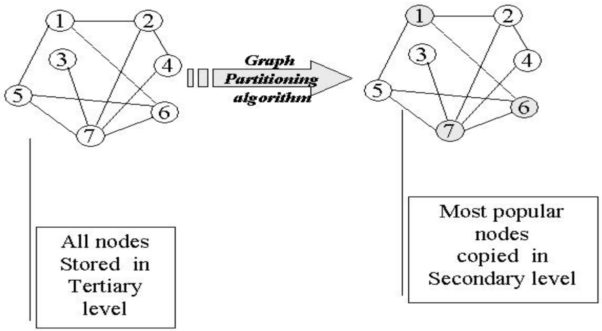

Fig. 2. Partiotioning browsing graph and storage in different levels.

- *Secondary Storage* : this level is defined by a disk storage subsystem, which is a disk array consisting disks of the same type and configuration. The disk array is considered in order to increase storage availability and parallelism since disks work in parallel servicing different requests. Disks are characterized by smaller seek and service times that tertiary storage devices. This multiple disk configuration is expected to be quite beneficial for multimedia applications where certain timing presentation constraints arise.

Since multimedia objects are analyzed in a number of physical objects we have defined a “pool” of physical objects participating in the multimedia application as a whole. Each physical object appears only once in the pool while it can be played in more than one multimedia objects and more than once within the same multimedia object. To define the data that will be elevated from tertiary to secondary storage the initial representation of the multimedia application (as a whole) and the multimedia node (in separate) has to be considered (Figure 1). All of the O $PObj$s participating in the multimedia application navigational graph will be initially stored in tertiary storage. Then, the graph will be partitioned and the $PObj$s participating in the most popular $MObj$s will be elevated and stored in secondary level. Figure 2 depicts the proposed elevation process. Therefore, the initial browsing graph G will be partitioned into two subgraphs G_1 and G_2. A copy of the $PObj$s participating in the $MObj$s (nodes) of the subgraph with the higher popularity (suppose G_1) will be elevated to secondary storage level. In order to identify the higher popularity subgraph we initially transform the initial directed graph G to a new undirected weighted graph G^* (according to an idea presented in [3]). This graph is defined as follows :

Definition 2: The *Weighted Graph* G^* is an undirected graph $G^* = (V, E)$ where $V = \{1, 2, \cdots, M\}$ is a set of M nodes corresponding to the M $MObj$s involved in the multimedia application and E is a set of undirected edges. Each edge (e_i, e_j) in G^* has a weight w_{ij}, associated with it, evaluated by :

$$w_{ij} = f_i p_{ij} + f_j p_{ji}$$

Graph partitioning algorithms have been studied extensively in earlier research efforts [6, 11, 14] and the graph partitioning problem is known to be NP-complete and thus there is no optimal solution. The graph partitioning algorithm adopted in this paper is based on a greedy approach to partition the weighted graph G^* . The subgraph G_1 will be the higher popularity graph which will involve the nodes with higher access frequency (f_i). At each iteration step in the algorithm a single node is added to the G_1 partition. The node to be added is identified by evaluation of the selection criterion (SC) defined as follows:

Definition 3 : The *Selection Criterion - SC* of an umarked node i of the weighted graph G^* is evaluated by:

$$SC[i] = \sum_{j \epsilon G_1} w_{i,j}$$

The node with the highest SC is selected, and added to partition G_1. The selection process is repeated until the desired number of nodes is added to the partition G_1. There is also a need to define a termination criterion for the definition of G_1 and for this reason, we define the Partitioning Threshold (PT) parameter :

Definition 4 : The *Partitioning Threshold (PT)* ($0 \leq PT \leq 1$) is a ratio variable to define the percentage of the nodes of the initial browsing graph that can be elevated to the secondary storage level.

4 Secondary Level Data Placement

4.1 Placement over the disk array

Here, we propose appropriate placement techniques for placement at the secondary level in order to exploit the storage topology. Only one physically stored copy of the considered *PObj*s will be placed on the secondary level (irrespectively to the number of *MObj*s in which it belongs to and the number of times it is displayed within each node of G_1). The criterion to "guide" the allocation of the *PObj*s to the disk array is the popularity of *PObj*s (Definition 1). The basic idea is that the *PObj*s which most likely are to be synchronized in the same *MObj* should be stored in different disks of the disk array so that they can be retrieved in parallel. Here we define the following parameters which will identify the synchronized objects :

Definition 5 : The *Synchronization Function* between two distinct *PObj*s, namely $PObj_i$ and $PObj_j$, participating in a *MObj m*, is evaluated by :

$$sync(PObj_i, PObj_j/m) = \begin{cases} 1 \; if \; PObj_i, PObj_j \; are \; synchronized \; in \; node \; m \\ 0 \; otherwise \end{cases}$$

Definition 6 : The *Synchronization Parameter* between two distinct *PObj*s, namely $PObj_i$ and $PObj_j$, is defined by

$$SP(PObj_i, PObj_j) = \sum_{m=1}^{M} sync(PObj_i, PObj_j/m)$$

The algorithm used for this multimedia placement approach is described in more detail in Table 1.

```
ALLOCATION-TO-DISKS
$V$:              Set of Vertices of the Initial Graph
$D$:              Number of Disks of the Disk Array
$disk[1 \ldots D]$:   Array of sets of objects being allocated to each one
                  of the D disks constructing the disk array
**Initial Values**
**for** i=1 **to** $N$
  $disk[i] = 0$;
UnallocatedObjects=$N_1$;
CurrentDisk=0;
**Greedy Algorithm Approach**
**while** there are still unallocated objects **do**
  Choose a vertex $v \epsilon V_1$ with the highest frequency of access
        as a starting vertex
  **if**(CurrentDisk==D) **then** CurrentDisk=0;
          **else** CurrentDisk++;
  UnallocatedObjects- -;
  disk[CurrentDisk]=disk[CurrentDisk]$\bigcup v$;
  $V = V - v$;
  SynchroSet=$\oslash$;
  counter=0;
  **for** every object $v_i \epsilon V$
      **if**($sync(v_i, v) \neq 0$)
            SynchroSet=SynchroSet $\bigcup v_i$;
            counter++;
  **order** the objects of the SynchroSet wrt the number of times
  they are synchronized with $v$
  **for** i=1 **to** counter      **if**(CurrentDisk=D) **then** CurrentDisk=0;
               **else** CurrentDisk++;
      UnallocatedObjects- -;
      disk[CurrentDisk]=disk[CurrentDisk]$\bigcup$SynchroSet(i);
      $V = V$-SynchroSet(i);
```

Table 1. Assigment of Physical Objects to Disks.

4.2 Placement on each disk

The algorithm of Table 1 points out the *Physical Objects* that should be placed on each one of the considered disks. The proposed placement approaches for disk placement are the following:

- The *Constructive Placement approach*:
  Under this approach we employ the popular *organ-pipe* placement algorithm. This algorithm has been proven to be quite efficient in secondary storage level placement as well as in tertiary storage devices. A detailed description of constructive placement techniques in general and organ-pipe placement in specific are given in [3, 20, 21].

- The *Iterative Improvement Approach*:
  The iterative improvement placement is an idea presented in [3, 20, 21] and is based on the Simulated Annealing algorithm. A detailed description of the simulated annealing algorithm, that is a widely accepted optimization technique, is given in [8]. The basic idea of the algorithm as we have adapt it to the placement problem is to start with an initial placement scenario and repeatedly modify it in search for cost reduction, that leads up to the global minimum. In this paper, we define our cost criterion to be the expected service time that is evaluated by the formula:

$$ExpectedServiceTime = \sum_{i=1}^{PD} \sum_{j=1}^{PD} pop[i]pop[j](s_j s_{rate} + t_j t_{rate})$$

  where i,j refer to the current head location (i) towards the requested location (j). Notice that s_j and t_j are the number of bytes to search and transfer (respectively), while s_{rate} and t_{rate} are the search and transfer rates (respectively). Finally, PD refers to the total number of $PObj$s to be stored in this specific disk.

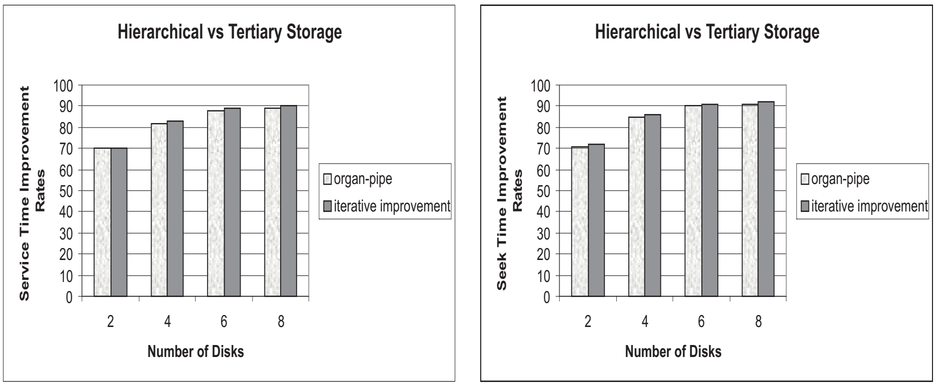

Fig. 3. Improvement rates in Service - Seek Times; Hierarchical vs Tertiary Storage.

5 Experimentation - Results

We have developed a simulation model for the hierarchical storage system described in Section 3, consisting of both tertiary and secondary storage models. Different placement strategies (organ-pipe, random, iterative improvement) have been employed and comparative results have been reported. For the experimental modeling the artificial workload of the multimedia objects been stored has been created as follows:

- The typical scenario is that the total number of physical objects of the pool increases with the number of nodes of the browsing graph. However, as the number of physical objects each node contains is uniformly distributed between 1 and the total number of physical objects (P) of the pool, we support larger size $MObj$s to better experiment with our simulation model.

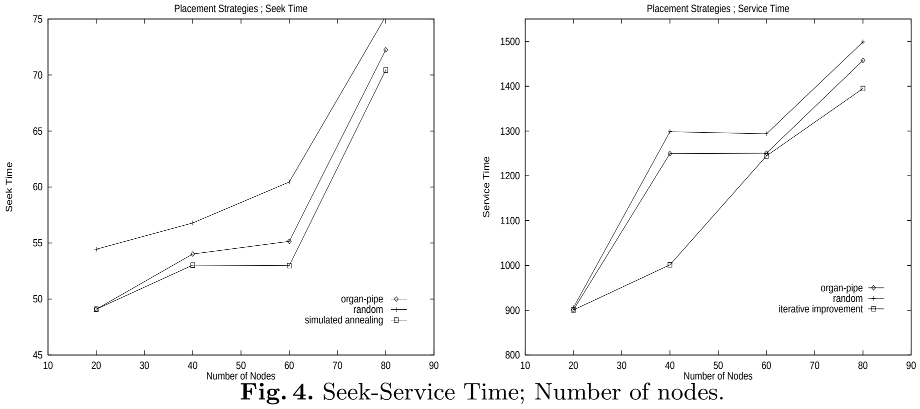

Fig. 4. Seek-Service Time; Number of nodes.

- each physical object size varies from some hundreds of KB to hundreds of MB to correspond the considered workload to real multimedia data sizes.
- a high percentage of the total tape space of the tertiary storage level is occupied. More specifically, for our tertiary storage system containing 4 tapes, 75% of the total available storage capacity is occupied.
- the percentage of the total disk space been occupied by the elevated physical objects remains also high. For example, this percentage reaches even 96% in the case of a disk array of two disks.

Figure 3 depicts the percentage of improvement for service and seek times (left and right part of this figure respectively), for the considered data elevation hierarchical storage approach under different placement strategies and for a varying number of disks of the disk array. This figure presents the comparison of the proposed elevation approach with respect to a typical placement at the tertiary storage level. The improvement rates are rather important and the proposed elevation is quite beneficial to the system's functionality and responsiveness.

Figure 4 presents the seek and service times obtained by the hierarchical storage configuration when different placement algorithms are implemented for the placement of data on the disk array. Organ-pipe and simulated annealing placement policies are proved to be better than random placement irrespective to the size of the multimedia application (denoted by the number of nodes of the x axis). The highest improvement rate of service (seek) time achieved by organ-pipe placement is 6.1% (8.6%) while the respective rates obtained by the iterative improvement algorithm are 25.2% (12%).

6 Conclusions - Future Work

This paper considers multimedia objects representation and storage and proposes a navigational multimedia model on which data elevation is employed under a multi-level hierarchical storage topology. The browsing graph structure is used to capture the users navigational pattern among the multimedia objects whereas a tree-like structure defines the relationships of the stored physical objects involved in a multimedia object. The simulation model developed is based on a hierarchical storage system of a tape library (tertiary level) and a disk array (secondary level) and both constructive and iterative improvement placement policies were considered for performing the storage. Experimentation results have indicated that iterative improvement is the most beneficial placement policy with significant improvement rates in both seek and service times and the considered elevation approach has been proven rather beneficial to the systems performance, while the data elevation process is considerably beneficial as compared to the one-level (tertiary) storage approach.

Further research should be employed in the area of combining prefetching and on-demand elevation under various data replacement algorithms, where the secondary level will be considered as a cache area for the tertiary storage level. Furthermore, different algorithms for disk data allocation to the disks of the disk array and data placement policies within each single disk can be applied in relation to techniques such as striping and replication.

References

1. Bertino E. and Ferrari E.: "Temporal Synchronization Models for Multimedia Data", *IEEE Transactions on Knowledge and Data Engineering*, Vol.10, No.4, 1998.
2. Brubeck D.W. and Rowe L.A.: "Hierarchical Storage Management in a Distributed VOD System", *IEEE 1996.*
3. Chen Y.T.: "Physical Storage Model for Interactive Multimedia Information Systems", PhD Thesis, Department of Electrical Engineering, Purdue University, 1993.
4. Chervenak A.L.: "Tertiary Storage - an Elevation of New Applications", PhD Dissertation, University of California at Berkley, 1994.
5. Christodoulakis S., Triantafillou P. and Zioga F.: "Principles of Optimally Placing Data in Tertiary Storage Libraries", *Proceedings 23rd VLDB Conference*, pp.236-245, Athens, Greece 1997.
6. Elsner U.: "Graph Partitioning - a Survey", Technische Universitat Chemnitz, December 1997.
7. Gibson G.A., Vitter J.S. and Wilkes J.: "Strategic Directions in Storage I/O Issues in Large-Scale Computing", *ACM Computing Surveys*, Vol.28, No.4, pp779-793, 1996.
8. Fleischer M.: "Simulated Annealing: Past, Present and Future", *Proceedings 1995 Winter Simulation Conference*, 1995.
9. Hillyer B.K. and Silberschatz A.: On the Modeling and Performance Characteristics of a Serpentine Tape, *Proceedings ACM SIGMOD Conference*, pp.170-179, 1996.
10. Ozden B., Rastogi R. and Silberschatz A.: "Research Issues in Multimedia Storage Servers", *ACM Computing Surveys*, Vol.27, No.4 , December 1995.
11. Pothen A., Simon H.D., Wang L. and Bernard S.T.: "Towards a Fast Implementation of Spectral Nested Dissection", *IEEE 1992.*
12. Prabhakar S. Agrawal D., El Abbadi A. and Singh A.: "Tertiary Storage: Current Status and Future Trends", Computer Science Department, University of California, Santa Barbara, CA, August 1996.
13. Ruemmler C. and Wilkes J.: "An Introduction to Disk Drive Modeling", *IEEE Computer*, 1994.
14. Schloegel K., Karypis G. and Kumar V.: "Graph Partitioning for High Performance Scientific Simulations", in *CRPC Parallel Computing Handbook* by Dongarra J., Foster I., Fox G., Kennedy K. and White A. (eds.), Morgan Kaufmann, 2000.

15. Sesardi S., Rotem D. and Segev A.: “Optimal Arrangements of Cartridges in Carousel Type Mass Storage Systems, *The Computer Journal*, Vol.37, No.10, pp.873-887, 1994.
16. Shriver E.: “Performance Modeling for Realistic Storage Devices”, Ph.D. thesis, Computer Science, New York University, 1997.
17. Triantafillou P., Christodoulakis S. and Georgiadis C.: “A Comprehensive Analytical Performance Model for Disk Devices Under Random Workloads”, *IEEE Transactions on Konwledge and Data Engineering*, 2001.
18. Triantafillou P. and Papadakis T.: “Continuous Data Block Placement and Elevation from Tertiary Storage in Hierarchical Storage Servers”, *Cluster Computing : The Journal of Networks, Software Tools and Applications*, 2001.
19. Triantafillou P. and Papadakis T.: “On Demand Data Elevation in a Hierarchical Multimedia Storage Server”, *Proceedings 23rd VLDB Conference*, 1997.
20. Vakali A., Terzi E. and Elmagarmid A.: “Representation Models for Video Data Storage”, *Journal of Applied System Studies*, Special Issue on Distributed Multimedia Systems with Applications, 2001.
21. Vakali A. and Terzi E.: “Video Data Storage Policies: an Access Frequency Based Approach”, *Computers and Electrical Engineering Journal*, 2001.
22. Vakali A. and Manolopoulos Y.: “Information Placement Policies in Tertiary Storage Systems” *Proceedings Hellenic Conference of New Information Technologies*, pp.205-214, Oct 1998.