A Two-Level Representation Model for Effective Video Data Storage

Athena Vakali *
Department of Informatics
Aristotle University
54006 Thessaloniki, Greece
avakali@csd.auth.gr

Evimaria Terzi
Department of Informatics
Aristotle University
54006 Thessaloniki, Greece
evimaria@csd.auth.gr

ABSTRACT

The main issues characterizing current video applications are their strong requirements for huge storage spaces and their need for timing synchronization. Video data storage is a critical research topic due to the so-called *I/O bottleneck* problem which affects the quality of service of video applications. This paper introduces a two level video data representation model in order to guide video data storage on a tertiary storage subsystem. A simulation model has been developed to evaluate different video placement strategies based on both *Constructive* and *Iterative Improvement* approaches. Experimentation has been carried out for the proposed placement approaches as well as for a typical random placement policy which serves as a comparison reference. Iterative Improvement placement has been proven to outperform the other considered video data placement approaches, in both seek and service times.

Index terms: *multimedia data storage, video data representation, video data placement algorithms.*

1. INTRODUCTION

A tape-based storage system has been cosidered as a reasonable solution to the problem of lowering the cost of storage and management of continuous data. Video data, as a typical example of continuous data, are characterized by timing relations and constraints imposed by users interactions. Thus, their storage on a tertiary level medium should be further investigated.

The most important design issues of a video storage server is to provide jitter-free video services as well as to promote utilization of the storage bandwidth to accommodate more users. In [4] the issues of placing continuous objects on a tape-based tertiary storage system and also the ensuing replacement and scheduling policies have been considered. A new performance model for a video storage server, which takes into account striping strategies, disk scheduling policies, data placement schemes, block sizes, buffer requirements and initial delay time simultaneously, is proposed in [23]. The configuration of a storage system that supports different video data types is studied in [15], whereas in [18, 19] the configuration of a system to support sharing for continuous media type is studied. Placement techniques and scheduling algorithms that guarantee continuous display of objects within a heterogeneous disk storage system are introduced and evaluated in [25].

Issues concerning tertiary storage devices and tape libraries in particular, have been pointed out in [6, 7, 16].The physical structure of multimedia data storage subsystems is described in [11]. It is interesting to note that tertiary systems have different and diverse performance factors not applicable to all technologies as indicated in [16]. Data placement has been studied for Tertiary Storage SubSystems, in [7, 17, 24]. Iterative improvement placement algorithms, and Simulated Annealing in particular, have also been implemented in [5]. A detailed description of the simulated annealing algorithm is given in [21] while its implementation on database systems has been discussed in [20].

Representation models for multimedia data have been introduced to represent the temporal relations among objects and specify the timing at which discrete events occur. In [2, 3] a classification of the representation models, based on the notion of time is presented. The main classes been discriminated are the *timeline*, the *interval-based* and the *constraint-based* models. Also, a number of different representation approaches have been also introduced in [2, 3, 8] and classified into three main categories: *Graph Models*, *Petri-Net models* and *Object-Oriented Models*. Based upon the Object-Oriented approach, the STORM (Structural and Temporal Object-oRiented Multimedia) DBMS proposed in [1] integrates structural and temporal aspects for managing different presentations of multimedia objects. In [10, 12] video objects representation models are categorized into *Stream-Based Models* and *Structured Models* with respect to their physical requirements and from the perspective of the DataBase Management System. In [22] a different multime-

*this research work has been supported by NSF grant MSI-9972883, while the author was a visitor at the Computer Science Dpt., Purdue University.

submitted for publication at the 6th Int. Workshop on Multimedia Information Systems, MIS 2000 Chicago, Illinois, USA

dia data representation model is proposed under a considered database schema based on a hierarchical tree structure for video objects representation. Furthermore, a new timeline model is proposed in [13] which captures user's interactivity on a set of multimedia documents.

This paper presents a model for video data placement on a considered tertiary storage subsystem, under a specific video data representation model. The paper's main contribution focuses on the following key issues :

- the *navigation path* among various video objects is considered for video data representation and their storage. The relationships among various video objects capture the user's access patterns on a considered multimedia storage server. These access patterns are considered to guide the storage policies and video data are no longer considered to be independently accessed.
- the *physical objects* corresponding to the video clips (included at a particular video data object) define the actual storage units involved in the data placement policies. Then, the frequency of access of a particular video clip specifies the popularity of the corresponding storage unit, such that the data placement policy will favor the most popular frequently accessed video clips.

The structure of the remainder of the paper is organized as follows. In Section 2 the proposed representation model is introduced. Section 3 provides a description of the data placement criteria and both Constructive and Iterative Improvement placement algorithms are further discussed. The main modules of the simulation and the storage model are given in Section 4. Experimentation and results are presented in Section 5 with discussion and result comments. Finally, conclusions and future work topics are discussed in Section 6.

2. VIDEO DATA REPRESENTATION MODEL

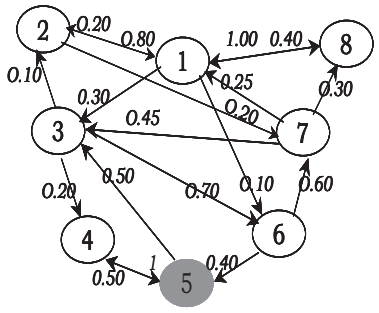

Figure 1: A Browsing Graph for an 8 video objects example.

A video application is based on the interaction and interconnection of video objects. For example, in a video movie application each user/client navigates (in an interactive way) through several video objects which correspond to a set of video clips. Video clips consist of video frames which are the last level of granularity in video structuring.

Definition 1 : *Video Clip* is a sequence of video frames. Each video clip has its own size, duration, presentation rate and is stored as an entity in files or blocks of the storage system.

Definition 2 : *Video object* is a set of video clips which are characterized by temporal constraints. A video object is defined as a set of tuples :

$$\text{Video Object} = \{(vc_1, s_1, e_1), (vc_2, s_2, e_2), \ldots, (vc_n, s_n, e_n)\}$$

where vc_i represents the identity of the i-th video clip belonging in the video object and s_i,e_i are the start and the ending times (respectively) for the i-th entity involved in a particular video object.

In our case a graph model has been considered to represent video data as well as the overall access pattern among different video objects. The basic idea of the proposed model is depicted in Figure 2. A user "moves" from one video object to another and this navigation can be represented as arcs in a directed graph. The nodes of the graph correspond to distinct video objects. For example, such an 8-node graph is depicted in Figure 1 for a set of eight distinct video objects. The model considers a further representation analysis for each of the nodes since each node (video object) consists of a number of storage objects (video clips). The structure of the internal node is represented by a tree like structure, based on the timeline representation model. The considered two representation levels are depicted in Figure 2 and they will be described below in further detail :

- **External level :** user's interaction can be represented by using the browsing graph as a "map" of nodes (video objects) visited by the user. Each node in the browsing graph corresponds to a video object itself, while the directed arcs represent the relationships among the various nodes (based on the idea of [5]).

  Definition 3 : *The Browsing Graph* is a directed graph $G = (N, A)$ where $N = \{1, 2, \cdots, k\}$ is a set of k nodes corresponding to k video objects and A is a set of directed edges connecting specific pairs of N. Additionally, every edge in A is weighted by an access or transition probability.

  Any Browsing Graph can be uniquely defined by the so-called transition matrix:

  Definition 4 : The *transition matrix* P associated with the graph G, is a $(k \times k)$ matrix of access or transition probabilities , where by p_{ij} $(i, j \epsilon \{1, 2, \cdots, k\})$ we denote the probability of accessing node j from node i at a single step.

  It is obvious that the proposed model is a homogeneous Markov chain since the transition probabilities are time-independent. The notion of the "system" in the Markov chains terminology stands for the user's actions while the "states" of the system are the video objects or the nodes of the graph. It is also clear that the transition matrix P is a stochastic matrix, i.e. its

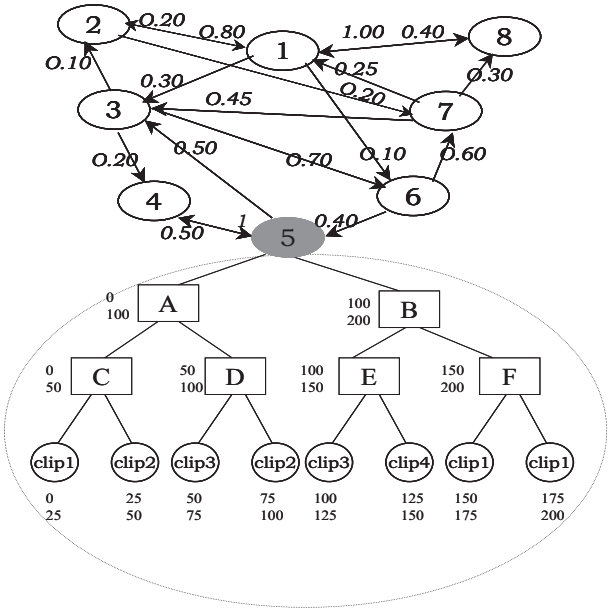

Figure 2: Graph-Tree representation

elements are either zero or positive and its row sums are all ones, that is $\sum_{j=1}^{k} p_{ij} = 1$ for all i.

Theorem 1 : If P is the transition matrix of a homogeneous ergodic Markov chain, then there is a unique vector $f = (f_1, \cdots, f_k)$, such that

$$\lim_{k \to \infty} P^k = \begin{pmatrix} f \\ f \\ \vdots \\ f \end{pmatrix} \qquad (1)$$

Proof : A thorough study and classification of finite Markov chains and the proof of this theorem is given in [9, 14]. The theorem gives us a way of approximately evaluating the access frequencies of the nodes, by simply calculating powers of the transition matrix. For example, if we raise the transition matrix of the Figure 2 examlple to 32-th power, the result will be a $k \times k$) matrix with equal rows, each row corresponfing to the vector f of access frequencies :
$f = (0.2397, 0.1080, 0.1627, 0.0325, 0.0877, 0.1379, 0.1043, 0.1272)$.

Definition 5 : The row vector $f = (f_1, \cdots, f_k)$, where $\sum_{i=1}^{k} f_i = 1$ and

$$fP = f \quad or \quad f_j = \sum_{i=1}^{k} f_i P_{ij}, \quad \text{for} \quad j = 1, 2, \cdots, k$$

is called *vector of access frequencies* of the video objects $1, \cdots, k$ and its elements provide metrics to identify the popularity of each video object involved in the browsing graph.

Theorem 1 gives a way to evaluate the relative frequency of accessing (retrieving) nodes $1, \cdots, k$ respectively in a long run, based on the transition probabilities of the initial browsing graph. It is known that in the theory of stochastic processes the vector f is called the equilibrium or stationary distribution of the Markov chain since any element represents the limiting probability of accessing the node i after infinite number of steps (in the long run).

- **Internal level :** At this level each node of the global view structure is further analyzed. Since timing relations and constraints arise among clips within a node, there is a need to focus on the node's content with respect to clips relationships over time and users' access pattern. Thus, a video object (i.e. a movie) can be considered as a collection of video clips. Each clip is considered to be the *storage object*, and such storage objects have variable sizes and durations. Furthermore, a clip may be part of more than one video objects. These storage objects are members of continuous media data and should be retrieved and displayed at a prespecified continuous rate. A timeline object oriented approach is introduced here in order to represent a single node of the external browsing graph (a similar approach has been presented in [13, 19]). The considered object oriented approach supports all the necessary temporal information needed for the synchronization constraints among the video clips in a single video object. The proposed model is a timeline model because the starting and ending times of the physical objects displays are defined with respect to the absolute time of the specific video object they belong to. As mentioned above, video objects are organized in a hierarchical tree structure. The introduced "time segment" tree consists of video clips (placed in leaves) and their timing relations (indicated in internal nodes). More specifically the model includes the following data structures :

  - *VC-ARRAY :* which is the video clip array with

its i-th element denoting the i-th video clip of the represented video object .

- *TIME-SEGMENT TREE* : The attributes of the Time-Segment Tree are the following :
  1. Each node represents a time sequence $[x, y)$, starting at time unit x and including all time units, but not time unit y.
  2. The leftmost leaf denotes the time interval $[z_1, z_2)$, the second from left $[z_2, z_3)$, the third $[z_3, z_4)$, and so on. Node N with two children representing the intervals $[p_1, p_2)$,$[p_2, p_3)$, represents the $[p_1, p_3)$,
  3. Every leaf of the tree is associated with a video clip that is displayed during the interval denoted by the specific leaf.

Example 1 : Table 1 represents the VC-ARRAY from which the Time-Segment Tree of Figure 2 is derived.

Video Clip	Time segment
VC-1	0-25
VC-1	150-200
VC-2	25-50
VC-2	75-100
VC-3	50-75
VC-3	100-125
VC-4	125-150

Table 1: Time-segment table example

The above described (two level) model will be referred to as *Graph-Tree* representation model. Thus the *Graph-Tree* representation model refers to a two-level representation model. A browsing graph is used to represent data at the external level while a hierarchical tree structure is adopted as an internal representation scheme.

3. VIDEO DATA PLACEMENT ALGORITHMS

3.1 The Placement Criteria

Certain criteria are necessary to guide the placement of video clips in order to propose effective video data physical layout. As mentioned in the previous section, the video objects include a number of video clips which specifies a "pool" of physical objects. However, only one copy of each video clip is kept in the storage system. Each video clip should be placed effectively and appropriately to guarantee high quality of service.

Definition 6 : The *popularity of a video clip x* in the Graph-Tree representation model is defined by

$$pop[x] = \sum_{i=1}^{k} f_i \times (number\ of\ object\ x\ playouts\ in\ node\ i)$$

where f_i is the frequency of access (Definition 5) of the video object corresponding to node i estimated by the formula given in Theorem 1. It is obvious that the higher the frequency of access of a video the more popular this clip is. Notice that the popularity value of a video clip is higher when the clip is part of a "popular" node and when it should be played for several times within this node. Therefore, all objects included in popular nodes have a high popularity value.

The popularity of video clips within the Graph-Tree representation model is the basic criterion which will be used to guide the implementation of the data placement algorithms on a tertiary storage subsystem. Tertiary storage systems and tape libraries in particular, are quite appropriate to accommodate voluminous video data as they are characterized by high storage capacity. Furthermore, recent technological advances have reduced the seek and service times of these devices and thus they can be seriously considered as an active part of the storage hierarchy even in the case of video data where certain timing constraints are imposed. Our data placement problem is to store C video clips onto T tapes. We perform placement on tapes with Z zones The considered data placement algorithms fall into two main categories : *Constructive placement* and *Iterative improvement placement*.

3.2 Constructive Placement

```
pool[1..C]        // Pool of C video clips
pop[1..C]         // Popularity of the C video clips
notstored []      //  array index to possible non-stored clips

VC[1..C] ← sorted  pool [ ] array in decreasing pop[ ] values

i ← 1
j←1
while (there are still Unallocated Clips and free Tape Space )
{
   STORE i-th clip  in  (i mod T) -th tape at first available segment.
              Estimated be the  Organ-pipe   (or camel) placement
   if (the clip is stored)
      i ← i+1
   else if (space is not enough)
       STORE i-th clip  on the  next tape with adequate free space.
       if (the clip is stored)
              i ← i+1
       else   // there is no tape with enough space available
              notstored[j]← i
              j ← j+1
              i ← i+1
}
```

Figure 3: Constructive Placement Algorithms

The *organ-pipe* and the *camel* placement algorithms are considered as indicative policies under the Constructive placement approach. Figure 3 presents these algorithms as applied in a tape library storage system with T tapes. The organ-pipe and camel placement policies as implemented within a tape consisting of Z zones are summarized as follows :

```
C0 : Starting condition value;
R:Reduction Value for the condition( 0 <R < 1)
N : Number of perturbations (if no improvement of Pbest occurs )

Pini ← initial Placement (randomly selected) ;
C = C0 ; P = Pini ; Pbest = Pini ;
Repeat while STOP = False ( Conditional loop )
        STOP = True ;   POINTER = 1
        Repeat while  POINTER ≤ N ( Perturbation loop )
                Ptry  =  P
                cost = cost(P )
                Ptry = perturb ( P )
                Dcost = cost ( Ptry ) - cost ( P )
                if (Dcost < 0 ) then
                        P = Ptry  ( accept the improvement )
                        STOP =  False
                else
                        p = exp ( - Dcost/C )
                        u ← random number in U( 0,1 )
                        if (u < p ) then
                                P =  Ptry  ( accept the wosrening )
                                STOP =  False
                        end if
                end if
                if ( cost (Ptry ) < cost ( Pbest ) ) then
                        POINTER = 1
                        Pbest  =  Ptry
                else
                        POINTER ++
                end if
        end repeat
        if  (STOP == False )
                C = C · R
        end if
end repeat
```

Figure 4: General Concept of Iterative Improvement algorithm

Organ-Pipe Placement

```
(A) Place the most popular object
    on the middle zone (Z/2) of the tape
(B) Allocate the next two popular objects
    on either side of the middle zone
(C) Go to (B) until all objects are placed.
```

Camel Placement

```
(A) Divide the tape into two consecutive tapes
    consisting of (Z/2) zones
(B) Implement Organ-Pipe placement alternatively
    on the two consecutive tapes
(C) Go to (B) until all objects are placed.
```

3.3 Iterative Improvement

The iterative improvement placement, commences with an initial placement determined by a constructive placement procedure and is repeatedly modified in search for cost reduction. This algorithm is based on the *Simulated Annealing* algorithm which is a popular algorithm for combinatorial optimization used in our placement problem. The iterative improvement starts with an initial placement determined by a constructive placement procedure and is repeatedly modified in search for cost reduction. A description of the algorithm is presented in Figure 4. Next we comment on the modules such as the *initial placement strategy, the perturbation procedure* and the *cost function* :

- The initial placement of the physical entities within the tapes can follow either one of the constructive placement methods described above, or a random policy.
- The *rearrangement (perturbation)* of the clips within the tapes can be applied following two strategies.
  1. Interchange : Select two clips of the current configuration randomly, and interchange their positions.
  2. Rotation : Make a left (or right) circular shift of current configuration.
- The *Expected Service Time* is chosen as an indicative measure of the cost of the placement indicated by the end of each perturbation, and is evaluated by the following formula :

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

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).

4. THE SIMULATION MODEL

Our simulation model consists of the three main modules (Figure 5) :

- *The Data Representation Module :* the browsing graph is constructed based on users access patterns. Representation is performed in both Internal and External levels and the access frequencies are evaluated by using the formula given in Theorem 1. The Video clips popularity is estimated by the formula given in Definition 6.
- *The Tertiary Storage System Module :*

  The tertiary storage subsystem used in our simulation is a tertiary storage library. Tape libraries come in many different sizes and configurations and despite their significant differences, they all contain : *Drives, Robot Arms* and *Tapes or Disks*. Consequently, they can all be modeled very similarly. Our library is thought to have one robot arm, which is assumed to be capable of moving between any tape stored in the library. The robot arms are involved in three operations :

  - *Pick and Place :* A pick operation refers to the robot picking up a tape from its storage space in the library and taking it into the drive.
  - *Move* : A move operation refers to the robot moving between different locations on the shelf.

  Although there is variation in the times required for these operations, they are generally modeled as constant times. This is acceptable because the variation is much smaller than the total time required to exchange tapes or disks on drives. Consequently:

$$RobotArm\_Service\_Time = Pick\_Time + Move\_Time + Put\_Time$$

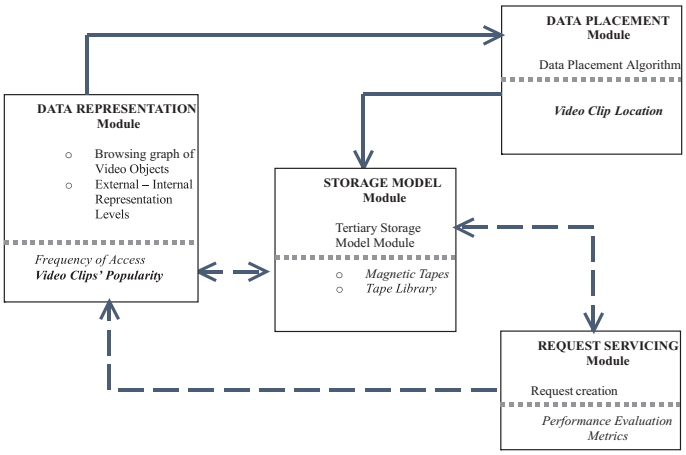

Figure 5: The Modules of the Simulation model.

The drives, which are also assumed to be identical, perform the following operations: seek, rewind, read, write, load and eject. The seek and rewind operations for tapes are modeled as constant startup times followed by a constant transfer rate. The tape access time is defined by the times involved in the servicing main actions :

$$Drive\_Service\_Time = Rewind\_T + Eject\_T + Load\_T + Seek\_T + Transfer\_T$$

Therefore, the total access time for a tape operation which includes a tape switch operation is defined as follows:

$$Total\_Service\_Time = RobotArm\_Service\_Time + Drive\_Service\_Time$$

The tapes in the considered tape library are considered as linear surfaces divided into a certain number of fixed-size *segments*, which are the smallest accessible parts of the tape. *Sections* consist of a number of consequent segments, while *tracks* consist of a number of consequent sections. The number of segments that will be allocated to a data object mainly depends on the object's and the segment's size. Thus the number of segments s reserved for a single stored clip is given by $s = \left\lceil \frac{size\ of\ object}{size\ of\ a\ segment} \right\rceil$.

- *The Data Placement Module :* one of the developed data placement algorithms (as described in Section 4) is applied under a selected tape topology and the location of each physical object is identified.

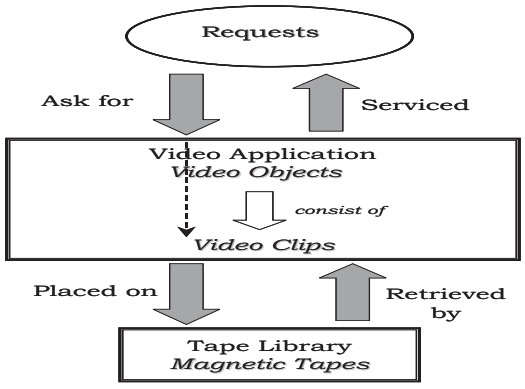

Figure 6: The request servicing process.

- *The Request Servicing Module :* The request workload refers to specific nodes of the external browsing graph. When a request arrives the video clips that correspond to the video object been pointed by the user are retrieved from the tapes on which they are stored. The clips are elevated on the cache memory and the request is serviced. The servicing procedure is also shown in Figure 6.

5. EXPERIMENTATION - RESULTS

We have run experimentation workloads for Zoned tapes as long as they are widely used and they tend to replace PBOT tapes. Numerous sets of requests were generated in order to evaluate the previously described placement schemes. The artificial workload of the video objects been stored has been

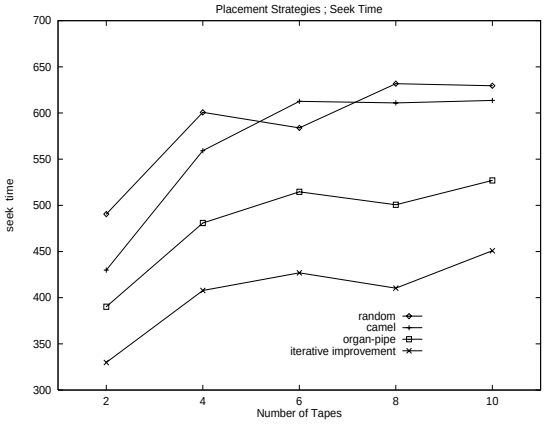

Figure 7: Seek Time ; Number Of Tapes.

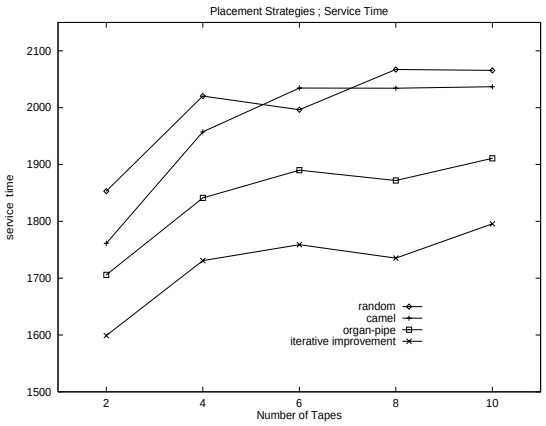

Figure 8: Service Time ; Number Of Tapes.

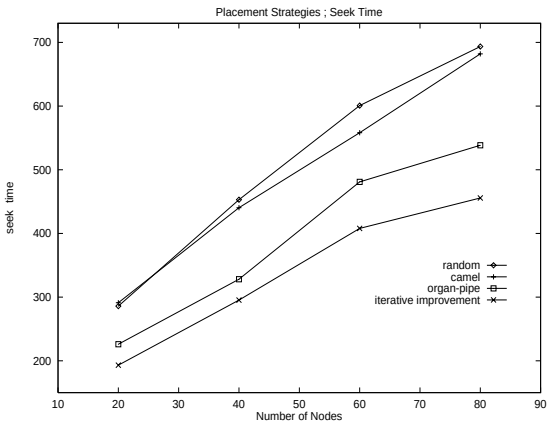

Figure 9: Seek Time ; Number Of Nodes.

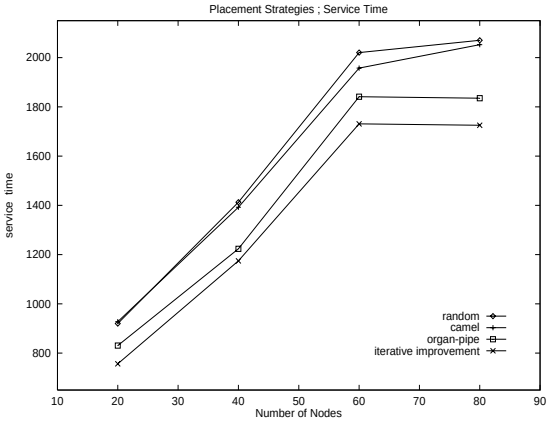

Figure 10: Service Time ; Number Of Nodes.

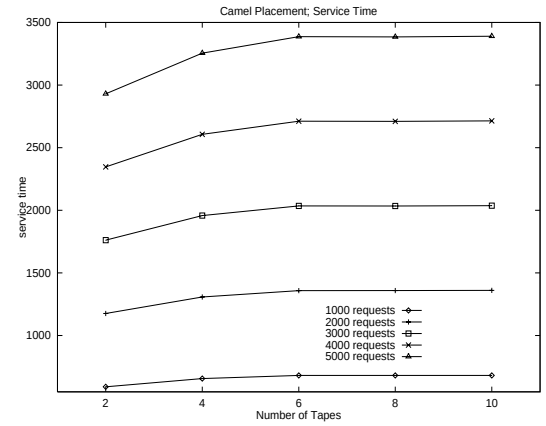

Figure 11: Camel Placement ; Number Of Tapes.

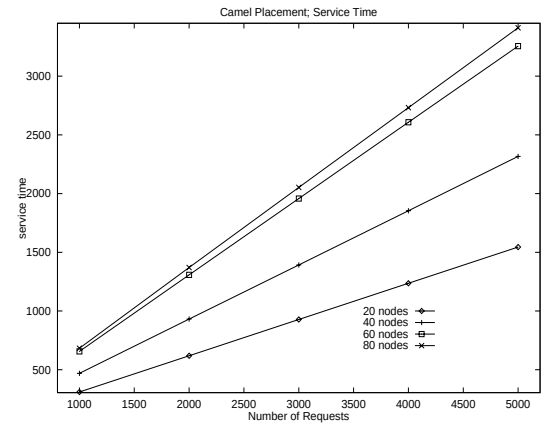

Figure 12: Camel Placement ; Number Of Nodes.

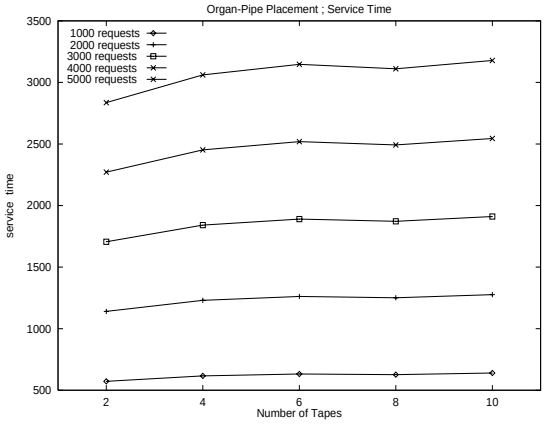

Figure 13: Organ-Pipe; Number Of Tapes.

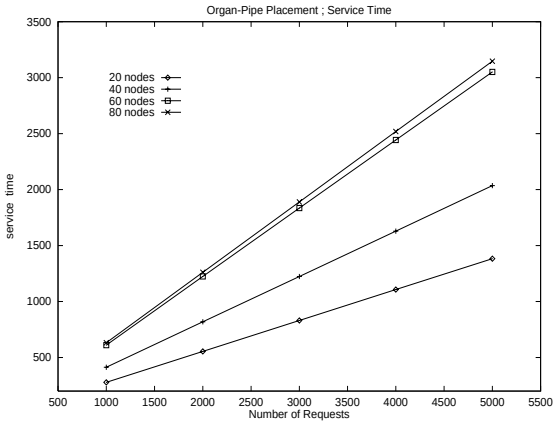

Figure 14: Organ-Pipe; Number Of Nodes.

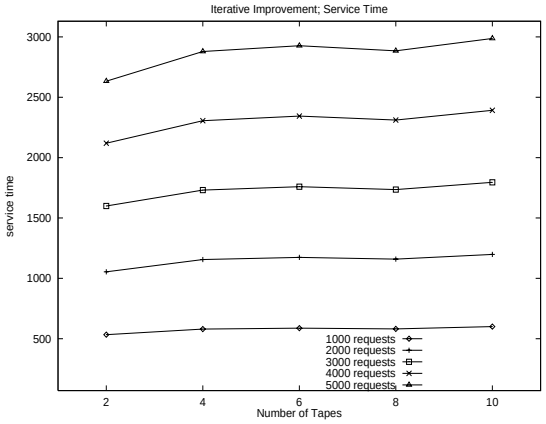

Figure 15: Iterative Improvement; Number Of Tapes.

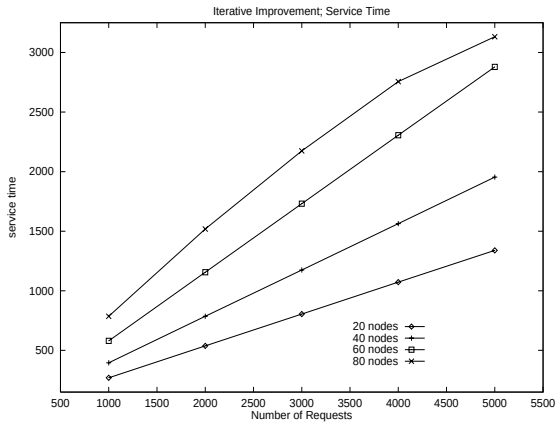

Figure 16: Iterative Improvement; Number Of Nodes.

	Camel Placement		Organ-pipe Placement		Iterative Improvement	
	service time	seek time	service time	seek time	service time	seek time
20 nodes	-0.8%	-2%	10%	20%	18%	32%
40 nodes	1.4%	2%	13%	27%	17%	35%
60 nodes	3%	7%	9%	20%	14%	32%
80 nodes	0.8%	2%	11%	22%	17%	34%

Table 2: Service-Seek time improvement rates between proposed policies and random placement.

	Camel Placement		Organ-pipe Placement		Iterative Improvement	
	service time	seek time	service time	seek time	service time	seek time
2 tapes	5%	12.4%	8%	20%	14%	33%
4 tapes	3%	7%	9%	20%	14%	32%
6 tapes	-2%	-5%	5%	12%	12%	27%
8 tapes	1.5%	3%	9%	21%	16%	35%
10 tapes	1.5%	2.5%	13%	16%	13%	28%

Table 3: Service-Seek time improvement rates between proposed policies and random placement.

created based on the following criteria :

- the total number of clips of the pool increases with the number of nodes of the browsing graph;
- the number of clips each node contains is uniformly distributed between 1 and the total number of clips in the pool;
- each clip's size varies from some hundreds of KB to hundreds of MB.
- it is obvious that the total size of video objects is equivalent to the size of real video data.

Furthermore, the workload was generated such that a large percentage of the total tape space to be occupied.

Simulation results refer to both seek and service time. More specifically, random ,organ pipe, camel and iterative improvement placement strategies have been implemented and the system's performance has been estimated for a system with a varying number of tapes $(2, \cdots, 10)$ and a constant number of nodes of the external browsing graph and vice versa (Figures 7, 8, 9 and 10). For storage systems with a small number of tapes (e.g. 2, 4 tapes) 75-90% of the total available storage capacity is occupied. This percentage inevitably decreases when the number of tapes increases, when the workload remains constant. This approach allowed us to experiment on the system's performance when the stored objects are either scattered among the available tapes or stored close to each other on a small number of tapes.

Results indicate that iterative improvement considerably improves system's performance. Figures 9 and 10 depict that as the number of nodes of the external graph increases the performance metrics' values become higher. More multimedia objects mean more physical objects being stored and therefore longer seeks in the storage system in order to the clips to be elevated on cache memory. We notice that organ pipe placement scheme proves to be better than both camel and random placement. Camel placement does not show any significant improvement on the overall system's performance. Figures 11, 12, 13, 14, 15, 16 show the system's performance when organ-pipe, camel and iterative improvement placement policies are implemented. These figures refer to the expected service time for systems with varying number of tapes $(2 \cdots 10)$ and multimedia applications with constant number of External Graph's nodes and vice versa. Table 2 summarizes on the improvement percentages depicted in Figures 9 and 10 as resulted for service and seek times under the different storage policies, compared to the random placement. Finally, a detail comparison of the placement algorithms behavior for a varying number of tapes as it is shown in Figures 7 and 8 is summarized in Table 3.

6. FUTURE WORK

In this paper a two level (Graph-Tree) video data representation model has been introduced and based on this model we have adopted certain criteria that guided the placement of data on a tertiary storage system. Experimentation concerning both constructive placement and iterative improvement placement algorithms indicate that iterative improvement considerably improves system's performance. Organ pipe placement scheme proves to be better than both camel and random placement. Camel placement has not been proven to be beneficial and it has even resulted in worse seek and service times that random placement.

Further research should extend the video data representation models so as to meet the demands of specific video applications, while adopting different criteria to guide video data layout in the overall storage system. Furthermore, information placement algorithms should also be implemented for other types of Tertiary and Secondary storage systems, including optical and magnetic disks. Moreover, we could extend our model in order to exploit all levels of the storage hierarchy in order to both improve response/service times. For example, we can propose a data placement approach based on objects access frequencies and dependencies, in order to "split" the browsing graph among secondary and tertiary storage levels. Thus, in our future hierarchical storage approach, secondary storage level could serve as a cache for the tertiary level. All objects will be stored in Tertiary Storage(TS) initially according to the placement policies mentioned above.

7. REFERENCES

[1] Michel Adiba : STORM : An Object-Oriented Multimedia DBMS, Chapter 3 of *Multimedia Database Systems- Design and Implementation Strategies*, Kluwer Academic Publishers.

[2] Elisa Bertino and Elena Ferrari : Temporal Synchronization Models for Multimedia Data, *IEEE Transactions on Knowledge and Data Engineering*, Vol.10, No.4, 1998.

[3] Yong-Moo Kwon,Elena Ferrari, Elisa Bertino : Modeling spatio-temporal constraints for multimedia objects, *Knowledge and Data Engineering 30*, pp.217-238, 1999.

[4] Jihad Boulos, Kinji Ono : VOD Data Management an Tape-Based Storage Systems, *SPIE Conference on Multimedia Storage and Archiving Systems III*, Boston, Massachusetts, November 1998.

[5] Yie-Tarng Chen : Physical storage model for interactive multimedia information systems, PhD Thesis, Department of Electrical Engineering, Purdue University, 1993.

[6] A.L. Chervenak: Tertiary Storage-An Elevation of New Applications, PhD Dissertation, University of California at Berkley, 1994.

[7] Stavros Christodoulakis, Peter Triantafillou and Fenia Zioga : Principles of Optimally Placing Data in Tertiary Storage Libraries, *23rd International Conference of Very Large Databases (VLB) Athens 1997*, pp.236-245.

[8] Soon M. Chung : *Multimedia Information Storage and Management*, Kluwer Academic Publishers.

[9] D.R. Cox and H.D. Miller : *The Theory of stochastic processes*, Chapman and Hall, 1977.

[10] Martha L.Escobar-Molano, Shahram Gandeharizadeh and Douglas Ierardi : An Optimal Resource Scheduler for Continuous Display of Structured Video Objects. *IEEE Transactions on Knowledge and Data Engineering*. Vol.8, No 3, June 1996.

[11] J. Gemmel and S. Christodoulakis : Principles of Delay-Sensitive Multimedia Data Storage and Retrieval, *ACM Transactions on Information Systems*, Vol. 10, No.1, pp.51-90, 1992.

[12] Shahram Ghandeharizadeh : Stream-based Versus Structured Video Objects: Issues, Solutions, and Challenges.

[13] N.Hirzalla, Ben Falchuk, and Ahmed Karmouch : A Temporal Model for Interactive Multimedia Scenarios. *IEEE Multimedia* 2(3): pp.24-31, Fall 1995.

[14] D.L. Isaacson and R.W. Madsen : *Markov chains theory and applications* John Wiley and Sons, 1976.

[15] Jonathan C.L. Liu, Jenwei Hsieh and David H.C. Du : Performance of a Storage System for Supporting Different Video Types and Qualities. *IEEE Journal on Selected Areas in Communications*, Vol.14, Issue 7, Sept. 1996, pp. 1314-1331.

[16] Sunil Prabhakar, Divyakant Agrawal, Amr El Abbadi, Ambuj Singh: Tertiary Storage: Current Status and Future Trends, Computer Science Department, University of California, Santa Barbara, August 1996.

[17] S.Sesardi, D. Rotem and A. Segev : Optimal Arrangements of Cartridges in Carousel Type Mass Storage Systems, *The Computer Journal*, Vol.37, No.10, pp.873-887, 1994.

[18] Cyrus Shahabi and Shahram Ghandeharizadeh : Continuous Presentations Sharing Clips. *ACM Multimedia Systems* 3(2),May 1995.

[19] Cyrus Shahabi : Scheduling the Retrievals of Continuous Media Objects,PhD Thesis, Computer Science, University of Southern California, 1996.

[20] Kien A. Hua, S.D. Lang and Wen K. Lee : A decomposition-based simulated annealing technique for data clustering. *Proceedings of the 13th ACM symposium on Principles of database systems*, Minneapolis, USA, May 1994.

[21] Mark Fleischer : Simulated Annealing : Past, Present and Future. *Proceedings of the 1995 Winter Simulation Conference.*

[22] V.S. Subrahmanian : *Principles of Multimedia Database Systems*, Morgan Kaufmann Publishers Inc., 1997.

[23] Shiao-Li Tsao, Yueh-Min Huang, Jen-Wen Ding : Performance analysis of video storage server under initial delay bounds. *Journal of Systems Architecture* 46 (2000) pp.163-179.

[24] A.Vakali and Y.Manolopoulos: Information Placement Policies in Tertiary Storage Systems, Storage Models for Multimedia Object, *Proceedings of the Hellenic Conference of New Information technologies*, pp.205-214, Oct 1998.

[25] Roger Zimmermann : Continuous Media Placement and Scheduling in Heterogeneous Disk Storage Systems, PhD Thesis, Computer Science, University of Southern California, 1998.