Heuristic Solutions for a Mapping Problem in a TV-Anytime Server Network*

Xiaobo Zhou[1], Reinhard Lüling[1], Li Xie[2]
[1] Paderborn Center for Parallel Computing, University of Paderborn
Fürstenallee 11, D-33102 Paderborn, Germany
Email: {zbo, rl}@uni-paderborn.de
[2] Department of Computer Science, Nanjing University, P.R.China
Email: {xieli}@netra.nju.edu.cn

Abstract

This paper presents a novel broadband multimedia service called TV-Anytime. The basic idea of this service is to store broadcast media assets onto media server systems and allow clients to access these streams at any time. We propose a hierarchical structure of a distributed server network to support a high quality TV-anytime service. A key issue, how to map the media assets onto such a hierarchical server network is addressed and formalized as a combinatorial optimization problem. In order to solve this optimization problem, a set of heuristic solutions by use of a parallel simulated annealing library is proposed and verified by a set of benchmark instances. Finally, the TV Cache is presented as a prototype of a scalable TV-Anytime system.

1 Introduction

Distributed multimedia systems are constantly growing in popularity thanks also to the presence of the Internet [1] [2] [3] [4]. Whereas an Internet newspaper can be accessed at any time, the access to high bandwidth audio/video information (all of which we shall henceforth refer to in this paper as media assets) provided by broadcasting companies is strictly time dependent and synchronous. In order to access broadband media interactively and time-independently, one has to make intensive use of mirroring mechanisms. The problem with todays VCRs is that recording is not very comfortable and especially the amount of content that can be recorded is usually limited. Therefore, one alternative is to install server systems that store a huge amount of digitally broadcasted programs and make these content accessible for the clients.

*This research was partly supported by a grant from Siemens AG, München and Axcent Media AG, Paderborn in the framework of the project HiQoS - High Performance Multimedia Services with Quality of Service Guarantees.

Advances in high-speed broadband network technologies and the cascading ratio of price to storage capacity make it feasible to provide the content of todays broadcasters in a similar way as the content stored on the Internet: This means the content can be accessed interactively and time-independently. A service that aims to realize such features is called *TV-Anytime* [2].

2 A Hierarchical TV-Anytime Server Network

It is the idea of the TV-Anytime service to record broadcast media assets and to make them available independently from the time they are broadcasted. The full potential of a TV-Anytime service may be reached if the recording of media coming from broadcasters is combined with indexing information provided by the broadcasters and with profile information coming from the clients. If the broadcasters deliver the metadata for each broadcast event, and these metadata are matched with information about the clients preferences, a personal TV program can be set up for the client that delivers the favorite TV contents to the client in a time-independent and interactive way. In this form broadcast media and Internet media can be consumed in the same fashion.

First two commercial implementations of the TV-Anytime service are based on the online-encoding of analog TV signals and are available for the consumer-market in form of digital VCRs [3] [4]. However, the presented architectures have two main drawbacks:

- In comparison with the large amount of TV content that is broadcasted every second, the storage capacity of the systems is rather small. In a scenario where the client describes his personal profile for an automatic recording of media assets, it can be expected that the automatic mechanism will not perfectly match the client's preferences because the systems allow to store only a few media assets.
- Clearly, some clients' preference program may be the same, e.g., popular news clips, sports. Therefore, it would be a good strategy to store media assets onto a server in a way that these content are offered over the network to some clients simultaneously, minimizing the overall storage space consumption and required communication bandwidth in the multicasting way [5].

To be successful, the scalability and robustness of the server network are likely to become predominant issues. The problems above lead directly to our proposal of a hierarchical structure of a server network for a TV-Anytime system. As depicted in Figure 1, we connect a number of media servers by a backbone network and some nodes of the network are connected to a number of clients. Such a hierarchical server network can include a small system that is installed close to the clients as well as larger systems that are installed within a local network or a public broadband network to provide one media asset to a large number of clients. This hierarchical structure can scale to a very large network dynamically and is suitable for Internet/Intranet working. A client connected to a media server has direct and consequently cheap access to all media assets stored on this local server. It is also possible for the client to access media assets stored on remote servers via the backbone network, although this remote access is more expensive.

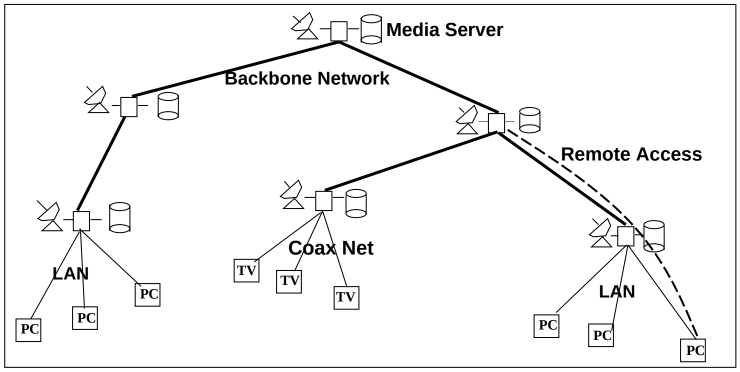

Figure 1: A hierarchical structure for TV-Anytime system.

3 The Media Mapping Problem

To implement a metropolitan TV-Anytime service on such a hierarchical server network, a new set of challenging research problems have been set forth. One of the key problems is how to map the media assets onto the server network and provide each requested media asset to the clients by use of the highest possible quality. The problem takes the access patterns and restrictions of the underlying hardware into account.

3.1 A Feature of the Mapping Problem

A very special feature of audio/video streams is that a media asset can be provided in different quality of service (QoS) which is determined by the encoding bit-rate of the media stream. Clearly, by use of a high encoding bit-rate, the media stream can be presented in good quality but also takes a large amount of storage capacity on the server and perhaps the communication load on the backbone network. Thus, the tradeoff that has to be solved by the mapping problem is to provide each client the selected media assets in the highest possible quality, taking the restrictions of the storage capacity of the media servers, the communication bandwidth of the backbone network, and the access patterns of the clients into account. This combinatorial problem can be formulated mathematically as a NP-complete optimization problem and can therefore be solved by use of some heuristic optimization solutions.

The mapping problem extends the File Allocation Problem (FAP) [6], a well known problem in parallel and distributed processing, using the nice feature of media assets that the assets are scalable in size as they are encoded in different bit-rates.

3.2 Formalizing the Mapping Problem

We assume the hierarchy of a server network is the binary tree with ring, i.e, all nodes in the leaf level of the tree are connected by a ring. All clients are connected to the leaf servers. The servers on upper levels are used as backup servers. Such a hierarchical structure for media server network can be modeled as a graph $N = (H, E)$, $H =$

$\{H_1, H_2\}$. H_1 is the set of backup servers which store assets with different bit-rates to provide them to leaf servers. H_2 is the set of leaf servers which all clients are connected with directly. The access patterns of clients connected to the leaf servers in H_2 is modeled as a graph $A = (V_A, E_A)$, and $V_A = H_2$, $E_A \subseteq \{(v, a) \mid v \in V_A, a \in M\}$. For a given server network N, an access pattern A, a storage capacity function c and a set of bandwidth B, now the question is how to map the media assets in such a way onto the server network that each access request can be fulfilled and totally the QoS is the best. An access request came from a client that is connecting with a leaf server v is fulfilled, if server v stores the requested media asset directly, or there is a path in the communication network to a backup server v' hosting the requested media asset and the communication path provides sufficient bandwidth to stream the asset to the requesting leaf v. Thus, the mapping problem can be formalized as follows:

Given: A server network $N = (H, E)$, $H = \{H_1, H_2\}$, access structure $A = (V_A, E_A)$ and a set of encoding bit-rates B.

Question: Is there a mapping π_a: $A \to P(E)$ and π_b: $A \to B$ with $\pi_a((v, a)) = \{(v, v_1), (v_1, v_2)\ , \ldots, (v_{l-1}, v_l)\}$ for a $l \in \mathcal{N}$ and $\pi_a((v, a)) = v_l$ and $\pi_b((v, a)) = b$ such that:

$\sum_{(v,a) \in A, \pi_a((v,a))=h} \pi_b((v, a)) \leq c(h), \forall h \in H$ and

$\sum_{(v,a) \in A, (u_1,u_2) \in \pi_a((v,a))} \pi_b((v, a)) \leq w_e(e), \forall e \in E, e = (u_1, u_2)$.

The optimization target is formalized as follows:

$QoS := \sum_{(v,a) \in E_A} \pi_b((v, a)) \to Max.$

4 Parallel Simulated Annealing Algorithms

Some heuristic methods, such as Simulated Annealing (SA) [7], have been shown to be very efficient for the solution of combinatorial optimization problems, e.g., TSP, but also for very specific problems in industrial applications, such as the vehicle routing problem. SA is based on a local search procedure. It defines a neighborhood relation on the set of all feasible solutions of a given optimization problem. Starting from an initial solution, a search process is performed on the network of feasible solutions, identifying one node of this graph as the current solution in each iteration.

For large instances of real-world optimization problem, even fast heuristics require a considerable amount of computational time. ParSA [8], a parallel library for simulated annealing, has been developed at University of Paderborn in order to speed up the computation. The generic decisions, such as the choice of the initial temperature, the cooling schedule, and stopping condition, are connected with parameters of the parSA algorithm itself. We need to think over the class of decisions which is problem specific and involved the space of feasible solutions, the cost function and the neighborhood function. Our intention is to parallelize our heuristic solutions with the parSA and therefore are able to reduce the computational time significantly.

4.1 Initial Solution

To compute an initial solution, each media asset a that is requested by clients connected with leaf server v has to be placed at least once on a server that lies on a path from v to the root of server network. Here, we propose a three-step initial solution algorithm.

According to the access pattern A, the first step of the initial solution algorithm places each requested media asset a, $(v, a) \in A$, with the minimal bit-rate on a server that is as close as possible to the leaf node v, because it is possible that no feasible solution will be found if the media assets can only be encoded with relatively high bit-rate. This makes sense for practical applications, since in this case each media asset can be provided to the clients at least in a low bit-rate quality. Then, in the second step the bit-rate of assets that have been mapped on the leaf servers is increased, if there is available storage capacity in leaf servers. In the third step of the algorithm, the bit-rate of media assets that have been mapped on the backup server is increased. The process starts at the backup servers that are nearest to the leaf servers.

4.2 Neighborhood Structure

4.2.1 Neighborhood Structure - Phase I

For a given feasible assignment of the media assets onto the hierarchical server network we compute a neighboring assignment in two phases.

In the first phase, a node in the hierarchical network of servers is identified. If the selected server v is a leaf server, an asset $a \in M$ and $(v, a) \in A$ will be chosen randomly. If the selected server v is a backup server, an asset a according to its *leaf children*'s access patterns will be chosen randomly. Then, the bit-rate of asset a is increased if it has been mapped onto the selected server, otherwise, it will be mapped onto the server. If above operations induce that the storage capacity of server v exceeds its limitation, the bit-rate of one or more media assets that have been mapped onto v will be decreased, which also perhaps induce the deletion of one or more media assets if they are encoded with the minimal bit-rate. We consider the assets that will decrease their encoding bit-rates or be deleted from the server v are chosen with *exponential distribution*, giving a higher priority to those assets which are mapped only with small bit-rates onto the server network. Therefore, the redundant copies which only have smaller bit-rates are deleted with higher probability. In experiments, we found these intelligent neighborhood structures leaded to better results.

4.2.2 Neighborhood Structure - Phase II

The perturbation of the current solution also effects other servers which access the media assets from the selected server. Thus, in the second phase, for all $(v, a) \in A$, if an asset a has not been mapped onto server v and therefore can not be accessed directly by the clients connected with the v, we have to find a path in the hierarchical network from v to a backup server v' which has been mapped a copy of a.

Ideally, the routing path is established to a backup server that stores a copy of requested asset a. This copy is encoded with a highest possible bit-rate. However, this greedy algorithm will get easily blocked for later requests because of violating the backbone bandwidth restrictions. On one hand each client hopes to get highest possible quality service. On the other hand, it is the case that each client hopes it can be given the minimal bit-rate quality if there is no enough bandwidth for its request. Therefore, the policy of selecting suitable backup servers might have to be based on the tradeoff between highest possible bit-rate and the total overhead onto communication bandwidth of backbone network.

Originally, we assumed asset migration only take place between servers that have a common path from the leaf server to the root which means there is only one path from selected backup server to the leaf server. However, in this case we found in some leaf servers there are some requests that can not be satisfied due to the communication load in their local routing trees while there are some idle communication bandwidth in other local routing trees. Since the network structure is the tree with ring, we allow asset migration to take place via the ring thus there are alternative routing paths. We proposed a backtracking routing algorithm to make the communication load in the tree evener and to induce more satisfied requests.

5 Performance Evaluation

To investigate in detail the performance of the algorithms for the mapping problem, we define a set of benchmark instances that reflect the implementation of large scale distributed media archives and their typical access patterns. Then we compare the Simulated Annealing solutions of the algorithms and the Upper bound of the solution.

Let M be the set of available media assets. The access pattern described by the graph $A = (V_A, E_A)$ is determined by a random process, which means each leaf's clients request a media asset $a \in M$ with a given probability p. In this way, we can identify a number of benchmark classes $R\_n\_m\_c_{Atom}\_w_e\_B_p$ given in table 1. The n represents the number of servers. The m represents the number of total media assets, thus $m =\mid M \mid$. The c_{Atom} (MB) represents the base storage capacity quantity. For a tree with k levels, we assume that the total storage capacity is $2^{k+1} \cdot c_{Atom}$. Each leaf node is given $3 \cdot c_{Atom}$, each backup node is given $1 \cdot c_{Atom}$ and the root is given $2 \cdot c_{Atom}$. The w_e represents the same communication bandwidth (MB) of each link in the tree network. Many studies in the literature dealing with service quality estimation of digitally coded video/audio sequences use a five-level scale for quality rating. For the set B, we assign the bit-rates from a minimum of 5MB/s (bad quality), 10MB/s (poor), 15MB/s (fair), 25MB/s (good), to a maximum of 40MB/s (excellent).

Benchmark Class	n	m	c_{Atom}	w_e	B	access pattern
$R\_7\_256\_500\_100\_\frac{1}{2}$	7	256	500	100	{5,10,15,25,40}	random $p = \frac{1}{2}$
$R\_15\_256\_750\_100\_\frac{1}{2}$	15	256	750	100	{5,10,15,25,40}	random $p = \frac{1}{2}$
$R\_15\_512\_750\_100\_\frac{1}{3}$	15	512	750	100	{5,10,15,25,40}	random $p = \frac{1}{3}$
$R\_15\_512\_1000\_150\_\frac{1}{3}$	15	512	1000	150	{5,10,15,25,40}	random $p = \frac{1}{3}$
$R\_31\_512\_750\_100\_\frac{1}{4}$	31	512	750	100	{5,10,15,25,40}	random $p = \frac{1}{4}$
$R\_31\_512\_1000\_150\_\frac{1}{4}$	31	512	1000	150	{5,10,15,25,40}	random $p = \frac{1}{4}$
$R\_31\_1024\_1000\_150\_\frac{1}{6}$	31	1024	1000	150	{5,10,15,25,40}	random $p = \frac{1}{6}$
$R\_31\_1024\_1250\_250\_\frac{1}{6}$	31	1024	1250	250	{5,10,15,25,40}	random $p = \frac{1}{6}$
$R\_63\_1024\_1000\_100\_\frac{1}{8}$	63	1024	1000	100	{5,10,15,25,40}	random $p = \frac{1}{8}$
$R\_63\_1024\_1250\_150\_\frac{1}{8}$	63	1024	1250	150	{5,10,15,25,40}	random $p = \frac{1}{8}$

Table 1: Definition of benchmark instances in a hierarchical server network.

We use parallel simulated annealing library (parSA) to test the set of benchmark instances defined above. The optimization target in cost function of parSA is the QoS defined in section 3.3. Thus, we compare the gap of parSA solution with Upper bound of QoS to verify the proposed heuristic solutions.

As simulated annealing is a stochastic method for the solution of the combinatorial optimization problem, we performed each run of the algorithm 10 times and took the average result. Table 2 shows the gap between the resulting QoS computed by the simulated annealing algorithms as well as the Upper bound of QoS for the benchmark instances. The measurement shows that the differences between the upper bound and the results gained by the parallel simulated annealing algorithms are very small, ranging from about 1.4 percent down to about 0.2 percent. It can be concluded that the algorithms can find good heuristic solutions.

Benchmark Class	$\frac{Upperbound - SAsolution}{Upperbound}$	$Average\ bit - rate$
$R\_7\_256\_750\_100\_\frac{1}{2}$	0.21%	17.57
$R\_15\_256\_750\_100\_\frac{1}{2}$	0.80%	18.81
$R\_15\_512\_750\_100\_\frac{1}{3}$	0.24%	13.86
$R\_15\_512\_1000\_150\_\frac{1}{3}$	1.41%	18.42
$R\_31\_512\_750\_100\_\frac{1}{4}$	1.08%	17.82
$R\_31\_512\_1000\_150\_\frac{1}{4}$	0.58%	24.05
$R\_31\_1024\_1000\_150\_\frac{1}{6}$	1.42%	18.21
$R\_31\_1024\_1200\_250\_\frac{1}{6}$	0.76%	22.35
$R\_63\_1024\_1000\_100\_\frac{1}{8}$	0.78%	26.60
$R\_63\_1024\_1250\_150\_\frac{1}{8}$	1.31%	33.21

Table 3: Performance of Neighbor_2 + backtracking routing Algorithm.

It can be seen from the average bit-rate of satisfied requests as depicted in Table 2, that most of the requests can be responded with *good* quality of service. It is expected that the average bit-rate of satisfied requests can be increased if extra storage capacity and communication bandwidth is added into the server network. For instance, the average bit-rate of benchmark $R\_15\_512\_750\_100\_\frac{1}{3}$ is 33% higher after 33% c_{Atom} and 50% w_e was added into the benchmark $R\_15\_512\_1000\_150\_\frac{1}{3}$. The same conclusion can be made from the comparison of $R\_31\_512\_750\_100\_\frac{1}{4}$ and $R\_31\_512\_1000\_150\_\frac{1}{4}$, $R\_31\_1024\_1000\_150\_\frac{1}{6}$ and $R\_31\_1024\_1200\_250\_\frac{1}{6}$, etc.

We also found that the difference between Upper bound of QoS and SA solution is mostly due to the wasted bandwidth in the backbone network. With a good bandwidth distribution strategy, we expect that the gap of SA solution and Upper bound of QoS will converge to 0.

6 TV Cache - A TV-Anytime System

In this paper, we proposed a set of heuristic solutions to solve the media asset mapping problem, a combinatorial optimization problem that arises in a hierarchical TV-Anytime system. The presented algorithms are combined with a parallel simulated annealing library (parSA) to test a set of benchmark instances. It is verified that the formalized optimization problem can be solved efficiently achieving near to optimal solutions in short time. In the optimal case, this parallel algorithms can be performed on the network of servers using the computational power that is available there.

The problem studied in this paper has a lot of practical relevance for the design and development of the prototype of a commercial TV-Anytime system, TV-Cache, which integrates Web technologies and the delivery of media streams in a seamless way.

The basis of the TV-Cache system is a commercial server system that performs MPEG streaming on the basis of a clustered PC architecture. The PCs run the Linux operating system. The media server can be used on a single PC, but also on a closely connected cluster of PCs, if a larger number of clients has to be supported. In the smallest configuration the system is used within the living room of a client. The client is connected to the Internet and to the inhouse antenna providing digital broadcast audio/video. Larger configurations are based on PC systems and are used as inhouse systems that feed the coax network of an apartment complex with TV-Anytime services or within a company or an ADSL network. Control information is transmitted via the Internet while media assets are transmitted by use of broadband technologies within the server network. This backbone connection allows to mirror media streams from one server to the others. Thus, the model and algorithms discussed in this paper are applied here.

References

[1] F.Cortes Gomez, Reinhard Lüling. *A Parallel Continuous Media Server for Internet Environments*. Proc. of International Conference on High-Performance Computing and Networking(HPCN Europe'98), Lecture Notes in Computer Science, 1998, pp.78-86.

[2] Reinhard Lüling. *Hierarchical Video-on-Demand Servers for TV-Anytime Services*, Proc. of 8th International Conference on Computer Communications and Networks (IC^3N), Boston, Massachusetts, IEEE Press, 1999, pp.110-117.

[3] ReplayTV: http://www.replay.com.

[4] Tivo: http://www.tivo.com.

[5] D.L.Eager, M.K.Vernon, J.Zahorjan. *Minimizing Bandwidth Requirements for On-Demand Data Delivery*. Tech.Report #4105, Computer Science Dept., University of Wisconsin - Madison, Aug. 1999.

[6] L.W.Dowdy, D.V.Foster. *Comparative Models of the File Assignment Problem*. Computing Surveys, Vol.14, No.2, 1982.

[7] S.Kirkpatrick, C.D.Gelatt, M.P.Vecchi. *Optimization by Simulated Annealing*. Science, Vol.220, No.4598, May 1983, pp.671-680.

[8] S.Tschoeke, G.Kliewer. *The parSA Parallel Simulated Annealing Library*. Technical Report, Department of Mathematics and Computer Science, University of Paderborn, http://www.uni-paderborn.de/∼parsa.