GMD Report

40

GMD –
Forschungszentrum
Informationstechnik
GmbH

Karl Aberer, Silvia Hollfelder

Resource Prediction and Admission Control for Interactive Video Browsing Scenarios Using Application Semantics

September 1998

Anschrift der Verfasser/Address of the authors:
Dr. Karl Aberer
Silvia Hollfelder
Institut für Integrierte Publikations- und Informationssysteme
GMD – Forschungszentrum Informationstechnik GmbH
Dolivostraße 15
D-64293 Darmstadt
E-mail: Karl.Aberer@gmd.de
        Silvia.Hollfelder@gmd.de

Abstract<superscript>*</superscript>

Highly interactive multimedia applications, like browsing in video databases, generate strongly varying loads on the media server during the presentation of media data. Existing admission control approaches for limiting the number of concurrent users and thus guaranteeing acceptable service quality are only suited for applications with uniform load characteristics like video-on-demand. We propose an approach to admission control that is based on the stochastical model of Continuous Time Markov Chains, which allows to describe the different presentation states occurring in the interactive access to the multimedia database. The model is derived from semantic information on the forthcoming browsing session, in particular, it considers the relevance of the videos to the user. In this way a more precise prediction on resource usage can be given for achieving the two goals of Quality of Service (QoS) and good server utilization. The admission control mechanism is part of a multimedia database architecture for supporting efficient browsing in large video collections.

Keywords: Admission Control, Interactive Multimedia Applications, Browsing, Multimedia DBMS

Zusammenfassung

Interaktive multimediale Anwendungen, wie beispielsweise das Browsen in Videodatanbanken, erzeugen während der Präsentation von Mediendaten stark schwankende Serverlasten. Die Aufgabe von Zulassungskontrollmechanismen ist die Begrenzung der gleichzeitig zu bedienenden Nutzer, um dadurch akzeptable Servicequalitäten zu erreichen. Bereits existierende Verfahren zur Zulassungskontrolle sind jedoch nur für Anwendungen mit relativ konstanten Lasten, wie beispielsweise Video-on-Demand Anwendungen, geeignet. In diesem Artikel wird ein Ansatz zur Zulassungskontrolle vorgestellt, der auf dem stochastischen Modell von sogenannten "Zeitkontinuierlichen Markov-Ketten" (Continuous Time Markov Chains) basiert. Dadurch wird die Modellierung verschiedener Präsentationszustände, die aufgrund des interaktiven Nutzerverhaltens entstehen, ermöglicht. Zur Modellierung werden semantische Informationen einer aktuellen Session verwendet, indem die Relevanzwerte einzelner Videos hinsichtlich einer vorherigen Anfrage berücksichtigt werden. Somit ist eine präzisere Vorhersage der Ressourcenverbrauchs möglich, wodurch sowohl Dienstgüteparameter (Quality of Service) als auch Serverauslastung verbessert werden können. Der Zulassungskontrollmechanismus ist Bestandteil einer multimedialen Datenbankarchitektur, die Video-Browsing in großen Datenbeständen unterstützt.

Keywords: Zulassungskontrolle, Interaktive Multimediale Anwendungen, Browsing, Multimedia DBMS

1 Introduction

Large digital collections of multimedia data, like Digital Libraries (DL), are getting increasingly important due to the widespread use of information networks like the World Wide Web. The amount of data available in digital multimedia collections is huge. Thus, a user needs to be supported to efficiently explore the digital collections by preselecting, browsing and presenting the data. Besides Digital Libraries, other applications also require this type of access, for example, previewing in pay-per-view systems or telelearning applications, where scholars from various disciplines like political scientists, psychologists, and historians, study moving images as primary source material [YYWL95].

In this article we focus on the problem of browsing multimedia data collections, in particular video collections. Browsing in video collections is particularly relevant, since content-based querying on video data is still not well supported. Browsing in video collections differs from other types of accesses to video databases, in particular video-on-demand applications. Only the relevant parts of the video are accessed and no complete videos need to be streamed to the user. By limiting data delivery to the relevant portions of videos the system throughput can be improved [CT95]. Frequent user interactions in browsing scenarios, like video selection and use of VCR-functions, and videos with different data rates, e.g., due to different encoding formats, cause highly varying data consumption rates. In addition, the required Quality of Service (QoS) may vary for different requests [SN95].

Thus, the media storage components have to provide mechanisms which are able to deal with this characteristics of highly interactive multimedia applications. In order to achieve the required presentation quality, the clients compete for limited resources on the server. The basic strategies to deal with limited resources can be classified as optimistic or pessimistic ones.

With optimistic strategies all requests are served as well as possible (best effort). These strategies are typically used in client-pull architectures, where the client requests small chunks of media data during presentation. The data requests arrive aperiodically at the server [RVT96]. In case of user interactions the client changes its request behavior, e.g., it will request larger blocks of a media or send more frequent requests. The client-pull architecture is suitable for interactive applications with varying resource requirements since it is not assumed that the required data rate will be constant for the duration of a presentation. Bottlenecks are dealt with either by the server or clients with various strategies, e.g., by means of quality adaptation mechanisms at the client [HKR97] or at the server [TK96][TKC$^+$97].

With pessimistic strategies resource reservations are made at the server, in advance. An admission control mechanism usually checks at the server if enough resources are available for the adequate delivery of data to a new client. If there are enough resources available, the client is admitted and the resources are reserved for this client until the end of the presentation. For interactive applications with varying resource demands this means that the admission control mechanism has to make predictions on the future resource consumption of client. This prediction can either be based on the past resource usage of a client, or can be predicted by exploiting knowledge on the semantics of the request. Both approaches have their pros and cons. Predictions based on past system behavior do not require high-level understanding of the semantics of the request and truly reflect the actual system usage. Thus, as long as the users behave in a uniform way, this approach appears to be appropriate. On the other hand, the implicit heuristic assumption that user behaviour does not change may be inappropriate if opposite knowledge exists. Therefore, in situations in which knowledge on the forthcoming access behavior is avail-

able, it might be substantial to derive predictions from that, though, inevitably, many assumptions and heuristics might be involved in the prediction. We give a concrete example for illustration. If users request access to a multimedia database for unrestricted browsing, a uniform model of usage is appropriate and access can be granted if it can be derived from access statistics that sufficient resources are available. On the other hand, if users request access to a multimedia database to browse a pre-specified subset of data, e.g. given by the result of a retrieval request, this subset may bear certain characteristics which allow much more precise estimations of future resource usage. For example, it might occur, that only low quality videos have been selected, and thus resource consumption is substantially lower than in the general case where both low and high quality videos are accessed equally.

In a previous paper, we have introduced an admission control framework for client-pull architectures that exploited the past client request behavior as an indicator for its future behavior [HA98]. On the one hand, this approach is fully application-independent, on the other hand, it does not exploit available knowledge on the application semantics for improved estimations of expected resource requirements. In this paper, we will introduce an admission control framework which allows to make predictions for the resource demands for browsing sessions in multimedia databases based on the semantics of the request. We propose an admission control mechanism for browsing applications which uses models of user behavior in a browsing session. The model is based on information that is extracted from the set of browsing candidates selected by a preceding retrieval request. We assume that the starting point to a browsing session is given by a retrieval request. The result of the retrieval is a hit list with corresponding relevance values for each hit. From this information we derive a Continuous Time Markov Chain (CTMC) which stochastically models the presumable behavior of a user. From the CTMC, we can derive a stochastic prediction of the future resource consumption of the client. This prediction is then used as an admission criterion. Thus, admission to the clients is granted in session-oriented manner. We expect that these techniques are not only suitable for browsing scenarios but with some adaptations also applicable for other types of access to multimedia databases, in particular preorchestrated multimedia presentations.

The paper is structured as follows: We first introduce the browsing system architecture in Section 2. In Section 3, we model different types of browsing scenarios by using the CTMC model. In Section 4, we describe how resource predictions can be made on the basis of the CTMC models and how they are used as admission control criteria. We conclude the paper with related work, in Section 5, and remarks on the future research direction in Section 6.

2   System Architecture

In this chapter, we describe the system architecture that is under implementation for a MM-DBMS at GMD-IPSI. It is designed to support highly interactive browsing applications [THE98]. The system architecture supports (1) content-based access to data, (2) continuous media presentation by means of client-side buffering mechanism, and (3) admission control for highly interactive applications.

Our browsing system is based on a client/server architecture where the server is responsible for the storage of continuous and discrete data. The client is responsible for requesting data units from the server, for presenting them, and for handling interactions with the user. The choice of a client-pull architecture gives adequate support for browsing applications since user interactions

6

are reflected by modified client requests.

The browsing prototype consists of the following components: a Multimedia Database Management System (MM-DBMS) that is responsible for the storage and retrieval of meta data and media objects, a multimedia retrieval engine based on abductive logic, an admission control module to restrict the access to the limited resouces on the server and to schedule data requests, a client-side buffering mechanism for media data, and a user interface for query formulation and result presentation. Figure 1 displays the relationships of the different components. These are described now, in more detail.
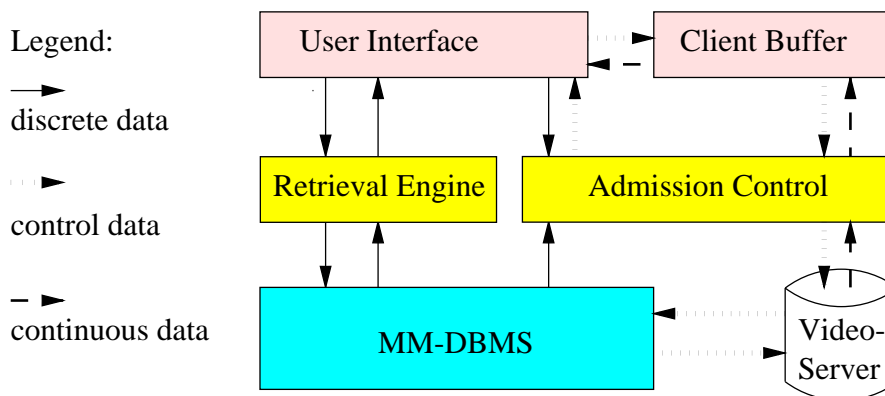
Figure 1: System architecture

2.1 Multimedia Database Management System

Our browsing prototype is implemented on the object-relational DBMS Informix Dynamic Server (IDS). The IDS enables the integration of so-called DataBlades which provide a flexible extension mechanism for new datatypes and their corresponding functions which are dynamically integrated into the DBMS's data model. We use the Video Foundation DataBlade [INF97] that forms the core of a system architecture for managing video data. It consists of three components: the IDS DBMS, the external storage manager, and the application program. The IDS DBMS stores the discrete data like text and images and the meta data for videos. It manages and controls the access to the external storage managers and devices. The external storage managers handle the storage of the various media streams. This means that the media data and meta data of the media are stored separately.

2.2 Retrieval Engine

The Retrieval Engine provides content-based access by employing different multimedia retrieval techniques, like feature extraction, feature aggregation, and classification for videos on scene granularity. Content-based access to media data is supported by conceptual queries. For example, when a user is interested in indoor shots he specifies "artificial light" and "artefacts". The queries are mapped by a rule-based engine to requests expressed in constraints on feature values [THE98]. As a query result, a hit list of stills, scenes, and videos, that have been found, is returned, together with relevance values - ranging between 0 and 1. The basic image features,

like edge analysis, grayscale, and entropy, are annotated in the IDS as meta data since feature extraction is a time consuming task. The features relate to single frames or scenes of a media. Rules define search criteria on the feature level which can be executed on the meta data [ME97]. The relevance values corresponding to a conceptual query are calculated by means of feature aggregation on video scene granularity. Since meta data management is important to the browsing application, a MM-DBMS implementation is best suited for providing the needed support [CT95].

2.3   Admission Control

The admission control module is located on top of the video server. It is responsible to schedule the limited server resources such as disk bandwidth and buffer space. Thus, given the delay-sensitivity of multimedia presentations, there is a limited number of clients that can be admitted for the service. The admission control module has access to meta data stored in the IDS DBMS.

In our architecture, an admission control module for highly interactive browsing applications is provided that considers the varying data rate requirements. Its tasks are divided into: (1) the admission of new clients, when it is assumed that system resources are sufficient, (2) the scheduling and adaptation of the single data requests of the admitted clients. In this paper, we will introduce an admission control mechanism that exploits knowledge about the query results, namely the content-based relevance of the single objects to a corresponding query.

2.4   Client Buffering

Since the Video Foundatation DataBlade does not support continuous presentation, we developed the Continuous Long Field DataBlade. It manages continuous data transport, client-side buffering, and client-side adaptation in distributed environments [HSHA98]. Additionally, we enhanced the client buffer strategy to support browsing applications by means of a content-based preloading and replacement strategy. The client buffer manager prefetches data by means of requesting single units of a continuous media from the server (client-pull). The client buffer-management strategy considers, in addition to the current presentation state, the relevance to a conceptual query result, too [EHT98]. The goal is to keep the most important scenes, corresponding to the current presentation state and to a previous query, in the buffer.

2.5   User Interface

The user interface enables the specification of a conceptual query that is sent to IDS DBMS and the selection of result scenes for presentation. At the server, the first hit request of a retrieval session is subject to admission control. During presentation the user has the possibility to control the presentation through VCR-interactions and to jump interactively to other hits.

3   Modeling of Browsing Applications

A major difficulty in estimating resource usage in interactive applications is the high variation of resource requirements. In this section, we will use a stochastical model, namely Continuous Time Markov Chains (CMTCs) to describe user interactions with a multimedia database server. It can be used to estimate future resource demands and, thus, to provide a more precise criterion

for an admission control mechanism. The admission control mechanism itself will be discussed in the subsequent section 4. We will use multimedia browsing sessions as an application scenario for inspecting multimedia retrieval results.

3.1 Modeling of Multimedia Sessions as Continuous Time Markov Chains

The retrieval and browsing system described in Section 2 delivers a result list L that contains references to scenes of videos or whole videos together with their relevance values as the result of a retrieval query. Thus an element $l_i \in L, i = 1, \ldots, |L|$ is of the form $l_i = < scene_i, rv_i >$, where $scene_i$ is an identifier for a video scene and $rv_i \in [0, 1]$ is a relevance value. Additionally, it is possible to compute physical information on the video scene from the meta data in the multimedia database, in particular, its duration $d(scene_i) \in R^+$ and the datarate $rate(scene_i) \in R^+$.

This information is available when a browsing session is started. The browsing session itself can be viewed as a state transition system, where the user switches between states for presenting particular videos and idle states for selecting the next video to be presented. For resource control it is important to consider, in addition to those states, the temporal dimension, i.e., the holding time of a state. A well established model to describe such state transitation systems stochastically are Continuous Time Markov Chains (CTMC) [Tij94].

A state transition process is specified by a CTMC by a set of states I, by holding times $\frac{1}{v_i}, i \in I$ and by transition probabilities $p_{i,j}$ with $i, j \in I, j \neq i$ and $\sum_{j \neq i} p_{i,j} = 1$ for all $i \in I$. If the systems jumps into state i, it stays in state i an exponentially distributed holding time with mean $\frac{1}{v_i}$ independently of how the system reached state i and how long it took to get there. If the system leaves state i, it jumps to state j with probability $p_{i,j}$ independently of the holding time of the state i. States are memory-less, which is called the Markovian property, i.e., the history how a state is reached is not relevant [Tij94]. Continuous time Markov chains are an extension of discrete time Markov chains, which do not model the holding times in the states.

Using CTMC for the modeling of a browsing session, the session states, i.e., the playback of a video scene or an idle time, are represented as corresponding states of a CTMC. The sojourn time or holding time in a state is the time until a user decides to change presentation process by an interaction. The transition probabilities denote the probability that a user switches from one session state to another one.

In our approach, we assume that the parameters determining a CTMC, i.e., the transition probabilities and the holding times of a state, are related to the relevance values rv_i of a hit $l_i \in L$. When a user finds a large number of hits he will not inspect all of them since the total presentation duration is too long. Typically, a user selects those scenes that have a high relevance with respect to the query. Furthermore, the time a user will spend to view a hit is dependent on its duration. The structure of the CTMC used to model the browsing session and the detailed relationship between the relevance values and the CTMC parameters are the subject of the next subsections.

3.2 Modeling Browsing Behavior by CTMCs

Depending on the application, a user may pursue different goals in a browsing session. Some users may aim at getting an overview of all hits in the hit list, others may intend to extract detailed information from the hit list. This results in different browsing behaviors. In the following, we

will discuss different possible browsing behaviors and model them by CTMCs. This discussion is not intended to exhaustively explore the issue of how browsing sessions are structured, but to illustrate how different assumptions on the nature of browsing sessions lead to structurally very different CTMC models. From this, we will eventually analyse the computational methods required for resource prediction. This discussion will be the basis for selecting appropriate and feasible resource predicition methods and, thus, eventually an admission criterion.

In the following discussion, we will first make a simplifying assumption on the viewing of the results. We will neglect different VCR-presentation states, like fast forward, fast rewind, and slow motion.

There are only two principle states we consider, namely the *idle states* in which the user selects the next scene to be presented, where no resources are consumed, and the *playback states* where particular videos are viewed, e.g., in standard playback mode. Transitions are possible only from idle states to playback states and from playback states to idle states. At the end of the section, we will then indicate how to additionally model different modes of presentation.

The structural differences in modelling browsing behavior by means of CTMC emerge from the different ways of how the browsing history is accounted for in the model. Since the CTMC itself is memory-less any use of historical information in the process needs to be encoded into different states.

3.3 Memory-free Browsing

In the simplest case, the selection of the next step is fully independent of the previous browsing steps. For modelling this situation it is sufficient to use one single idle state is and playback states $1, \ldots, |L|$ for the presentation of the different videos in the hit list. The transition probability $p_{is,i}$ is a function of rv_i only. In the simplest case, we choose the probabilities to be distributed in the same way as the relevance values, i.e., we use the *normalized relevance values* $\overline{rv}_i$ of a hit l_i, given by

$$\overline{rv}_i = \frac{rv_i}{\sum_{j=1}^{|L|} rv_j}, i = 1, \ldots, |L|$$

as transition probabilities. Then $p_{is,i} = \overline{rv}_i$ and $\sum_{i=1,\ldots,|L|} p_{is,i} = 1$, whereas always $p_{i,is} = 1$. An advanced model might use a weighting function in addition, e.g., to overproportionally increase the probability that videos of higher relevance are viewed.

For the holding times, we assume the following heuristic model: for short scenes, the mean of the exponentially distributed holding time is proportional to the length of the scene. There is a minimum presentation time d_{min} and for longer scenes the mean is limited by a maximal presentation duration $d_{min} + d_{max}$. In addition, we weight the mean by the relevance of the video, i.e. more relevant videos are viewed longer than less relevant ones. This heuristics is reflected in the following formula for the mean holding time:

$$1/v_i = d_{min} + d_{max} \frac{d(scene_i)}{d(scene_i) + d_{max}} rv_i, \ i = 1, \ldots, |L|$$

Example. To demonstrate the concepts, we use a running example in the following. A user query with 5 result scenes ($|L| = 5$) delivers the following results:

$$d(scene_1) = 5sec, rate(scene_1) = 1.5Mb/s, rv_1 = 0.8$$
$$d(scene_2) = 60sec, rate(scene_2) = 0.8Mb/s, rv_2 = 0.7$$
$$d(scene_3) = 20sec, rate(scene_3) = 4.0Mb/s, rv_3 = 0.7$$
$$d(scene_4) = 10sec, rate(scene_4) = 1.5Mb/s, rv_4 = 0.1$$
$$d(scene_5) = 20sec, rate(scene_5) = 4.0Mb/s, rv_5 = 0.05$$

By setting $d_{max} = 30sec$ and $d_{min} = 3$ we get the following (rounded) values for the holding times.

$$\frac{1}{v_1} = 6.4, \frac{1}{v_2} = 17, \frac{1}{v_3} = 11.4, \frac{1}{v_4} = 3.75, \frac{1}{v_5} = 3.6$$

For the idle state is, we assume a mean holding time

$$\frac{1}{v_{is} = 5}$$

which means it takes an average of 5 seconds to select the next presentation.

The transition probabilites in the example are then:

$$p_{is,1} = 0.34, p_{is,2} = 0.3, p_{is,3} = 0.3, p_{is,4} = 0.04, p_{is,5} = 0.02.$$

In Figure 2 the CTMC is given for the example. The numbers at the arrows represent the transition probabilities between the states.
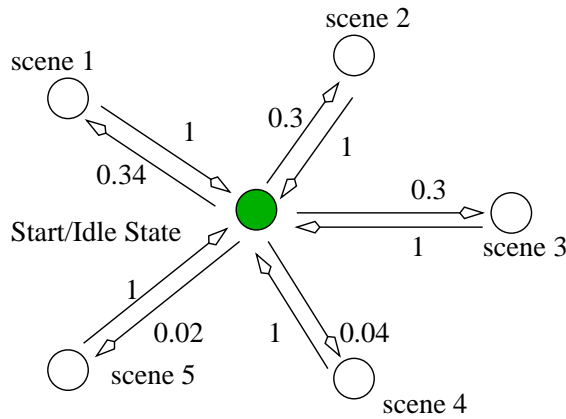

Figure 2: CTMC for memory-free browsing

3.4 General history-dependent Browsing

We introduce the most general model for browsing where the transition probabilities are fully dependent on the browsing history. In order to model this case we have to use a CTMC with the tree structure indicated in Figure 3. In this model, the root represents the start state, the nodes

at the first level represent all hits selected at first, the nodes at the second level all hits selected second, and so on. In this way, the CTMC represents all possible session histories. Each state represents a different viewing sequence of earlier videos and, since repetitions are possible, we end up with an infinite number of states. In contrast to the previous case, we have to distinguish a start state ss and different idle states is_h for each different presentation history h consisting of the sequence of videos that have been presented before.

We discuss now a simple model of how the transition probabilities can depend on the previous browsing history. A video that has just been viewed is not likely to be selected again. However, the longer a video has not been selected and the more other videos have been selected the more likely it becomes that the video will be selected again. Assume that the browsing session is in the idle state is_h belonging to a certain sequence of videos that have been selected before. Then for every scene $scene_i, i = 1, \ldots, |L|$ modify the relevance values as follows:

$$rv'_i = rv_i \frac{2n}{n + |L|},$$

where n is the number of times $scene_i$ has not been viewed in the history h. From the modified relevance values, we compute the normalized relevance values $\overline{rv'_i}$ and use them as transition probabilities. Note, that for $n = 0$ the we get $rv'_i = 0$ and for $n = |L|$ we get $rv'_i = rv_i$. The factor is monotonically increasing for $n > 0$.

Assume that in our running example $scene_1$ has just been viewed. Then the modified relevance values rv'_1 for $scene_1$ used to compute the transition probability for the consecutive steps are $0, 0.27, 0.45, 0.6, 0.71, 0.8, 0.87, 0.93 \ldots$ assuming the video is not selected within those steps. A fragment of the CTMC for this case with normalized $\overline{rv'_i}$ as transition probabilities is given in Figure 3. The dotted lines represent missing paths that are omitted due to the lack of space in the figure.
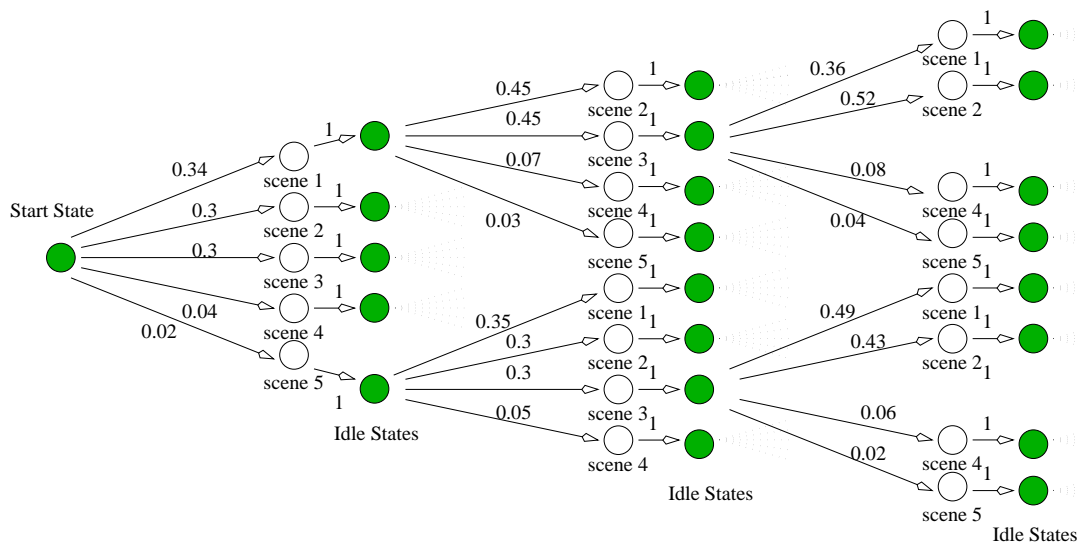

Figure 3: CTMC with tree structure for general history-dependent browsing

12

3.5 Browsing without Repetition

Up to now, we have assumed that the user is free to select any video for viewing an arbitrary number of times. We now investigate how further constraints on the selection of videos to be viewed impact the CTMC model for browsing.

The first additional constraint we consider is, that the user can view each video only once. Thus, we will obtain a finite CTMC as opposed to the previously discussed case of general, history-dependent browsing, where an infinite CTMC has become necessary. The general structure of the resulting CTMC is depicted in Figure 4. Since we assume that each video will be viewed only once, the number of subsequent states decreases in each level by the state that has been presented already and, thereby, the one-step transition probabilities increase for the remaining videos.
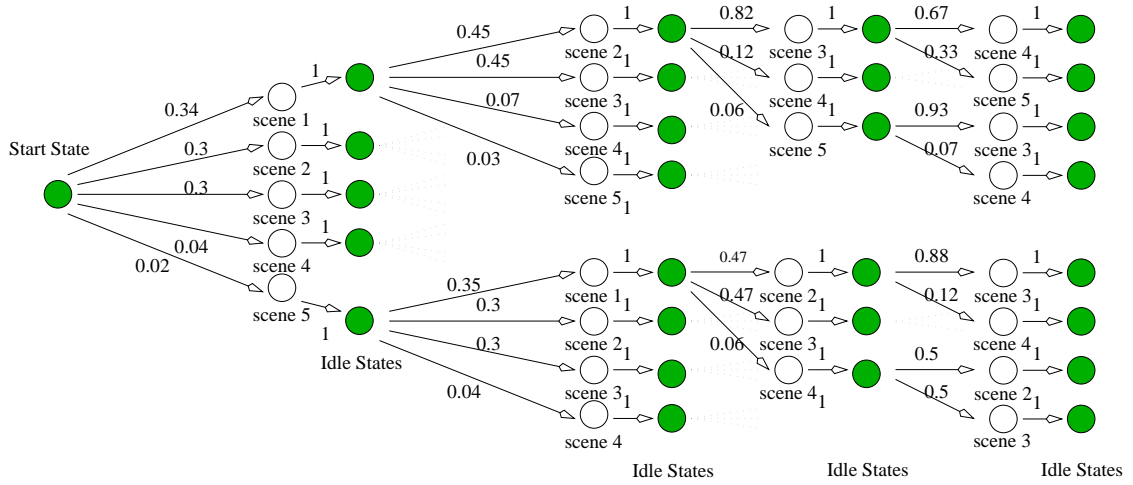

Figure 4: CTMC with tree structure for browsing without repetition

The transition probabilities are determined at each level from the normalized relevance values of the remaining videos. They are given for the case of our running example in Figure 4.

3.6 Browsing in Relevance Order

This is a very restricted form of browsing where the user can access the query result only in the order of their relevance values. Thus, the user basically only determines the holding times for each video that is viewed. In this case, we obtain a degenerated CTMC with a (nearly) linear structure, as displayed in Figure 5. The state on the left side represents the start state, the relevance values *rv* of the hits viewed in the playback states decrease from left to right. States with the same relevance are modeled by alternative state sequences that are accessed with the same probability.

Figure 5 displays the structure of the CTMC for browsing in relevance order for our running example.
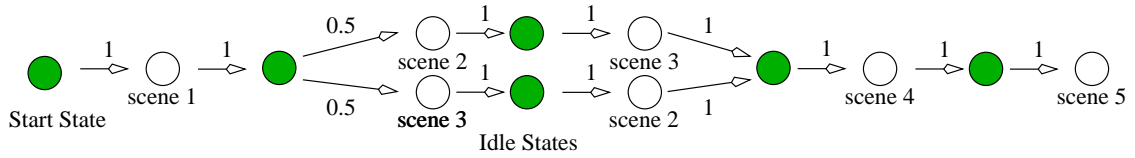
Figure 5: CTMC with sequential linear browsing

3.7 Representation of VCR-functionality

As indicated earlier, in the different CTMC models, we have used the simplifying assumption that the viewing of videos is represented by a single playback state in the CTMC. In practice, a user may use VCR-functionality. In the following we sketch how this can be accommodated within a CTMC model.

We assume now that, during the presentation of a video, a user may switch to a fast forward, fast rewind, or slow motion state. We determine that these interactions are only accessible from the playback state and extend thus a CTMC in the following way: for each video of the hit list, we introduce additional VCR-states for fast forward (ff), fast rewind (fr), and slow motion (sm). We assume fixed transition probabilities from the playback state to the VCR-states. The transition probability from the VCR-state to its playback state is always 1. The holding times can be derived from the holding times of the playback state, for example, by decreasing it for fast forward and fast rewind and increasing it for slow motion.

We illustrate the extended CTMC for our running example in the case of memory-free browsing. As transition probabilities to the VCR-states, we have arbitrarily chosen $p_{present,ff} = 0.25, p_{present,fr} = 0.05, p_{present,sm} = 0.1$ in this example. Figure 6 displays the corresponding CTMC when VCR-functionality is supported. Since the transition probabilities from the VCR-states to the present state are all equal to 1 they are not displayed in the figure.
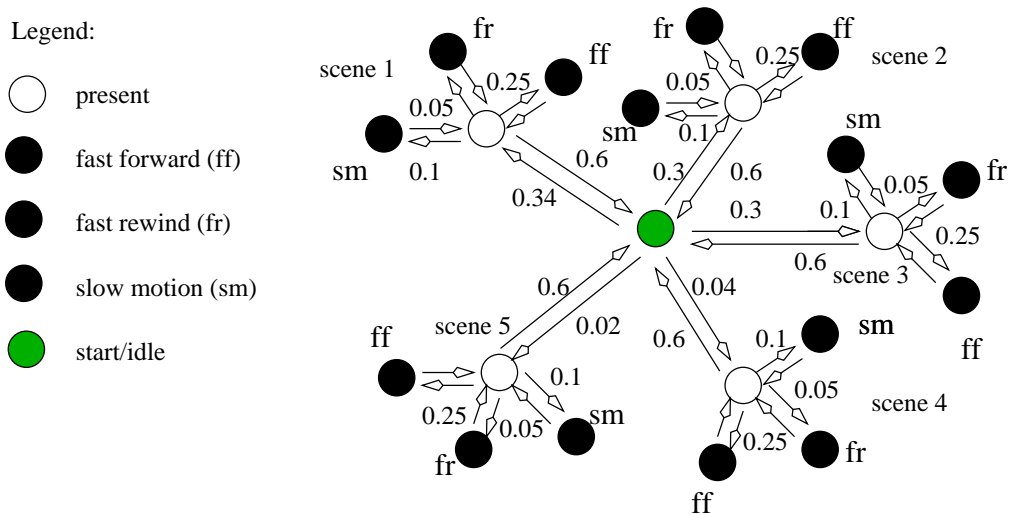

Figure 6: CTMC for memory-free browsing with VCR-functionality

14

3.8   Possible Refinements of the Modeling Approach

Appearently, in the modelling, a number of assumptions have been made on parameters and functions that capture certain characteristics of browsing sessions. It is beyond the scope of this paper to devise methods of how concrete parameters can be analytically derived from evaluations of concrete behavior of users. This is an important direction for future work. Such an approach allows not only to come up with better-substantiated heuristics on the user behavior, but would also allow to determine the necessary parameters individually for different users or application scenarios. As a drawback, individualized user parameters require additional bookkeeping mechanisms.

Some of the models introduced were of very high complexity. One can devise different ways of how this complexity could be reduced in order to obtain computationally more feasible models for browsing sessions, without giving up too much precision in the prediction. One obvious approach would be to aggregate states with similar characteristics, e.g., comparable resource consumption and holding time, and, thus, to substantially reduce the number of states in the model. Another possibility would be to allow for small errors and just omit less important scenes, i.e., those with low relevance.

In this paper, we have used a rather abstract view on how the occurence of a video within a query result is related to the probability that it will be accessed. Other parameters than the relevance value may influence the access probability to a video. In particular, the way in which the result list is presented at the user interface can play a substantial role. For example, for videos with the same or similar relevance value, the position in the result list can be of importance, or if the result hits are presented in a page oriented way, hits on the first page are more likely to be accessed than on later pages etc.. Thus the way, of how a multimedia presentation is generated from the hit list is certainly of relevance to the access behavior of the user. This discussion also shows that similar methods for modeling the user access to multimedia data may be employed for general preorchestrated multimedia presentations.

4   Admission Control Using Resource Prediction

4.1   Analysis of CTMC Models

In the previous section, we have modelled browsing behavior with CTMCs under various assumptions of how browsing might be performed. The main purpose of this analysis was to explore the question which structures occur in the CTMC models, how these structures are related to different assumptions on how browsing is performed, and how the resulting CTMCs are suited to compute a resource prediction.

The first important question in analyzing CTMCs is whether we have an open or a closed CTMC at hand. A CTMC is called closed if every state can be reached from every other state. This classification is important with regard to the applicable analysis methods. One distinguishes transient analysis and equilibrium analysis. Equilibrium analysis determines certain measures that are attached to a CTMC with regard to long term behavior. Transient analysis determines those measures over a given (short) finite time span. Equilibrium analysis is only applicable to closed CTMCs. With transient analysis, on the other hand we can also analyse CTMCs of infinite size, as they occured in the case of general memory-dependent browsing.

For the computational complexity of the analysis, the size of the CTMC is of importance.

Both, for the CTMC model for general history-dependent browsing and for browsing without repetition the size of the (relevant fragment of the) model grows exponentially in the number of hits, when a transient analysis is performed. For memory-less and sequential browsing the size of the CTMC is linear in the number of hits.

In this paper we will not be able to give a conclusive statement on which model and which type of analysis will prevail as the most relevant one. Rather, we will give the analysis for a selected case, namely memory-less browsing. As analysis method, we will use an equilibrium analysis. There are a number of reasons why this choice is reasonable and practical as well, in particular, considering the requirement that the resource predicition has to be performed efficiently:

From the computational viewpoint, encoding of histories into CTMCs leads to combinatorially explosive sizes of the resulting models and, thus, to prohibitively high costs in the analysis. In addition, equilibrium analysis is computationally simpler than transient analysis. A problem which further complicates transient analysis is the choice of the expected duration of the browsing session. This does not occur in equilibrium analysis.

When the impact of the history on the transition probabilities is small the equilibrium analysis is a good approximation of the transient analysis. This is also the case when only a few hits will be viewed in a browsing section, since then only a few transition probabilities change, too. In addition, there exists the possibility to redo the equilibrium analysis at a later stage with modified parameters and to accomodate changes that result from the previous history.

4.2 Resource Estimation for memory-less Browsing using Equilibrium Analysis

Since for each playback state the corresponding data rates are known, it is possible to stochastically determine an overall expected data rate for a single client session, based on its CTMC model for browsing. In the following, we will give the necessary steps to perform this calculation. Details on the mathematical background of this calculation can be found in [Tij94].

A closed Continuous Time Markov chain with bounded rates $v_i, i \in I$ has a unique equilibrium distribution $P_i, i \in I$, where the P_i can be interpreted as the probability that the Markov chain is in state i. In order to compute this equilibrium distribution one first transforms it into a discrete Markov chain by introducing so called transition rates $q_{i,j}$ with

$$q_{i,j} = v_i p_{i,j} \quad (*),$$

with $i, j \in I, j \neq i$. Based on the transition rates, the equilibrium distribution can be determined by solving the following system of linear equations, which has a unique solution.

$$v_i P_i = \sum_{k \neq i} q_{k,i} P_k, i \in I$$

$$\sum_{k \in I} P_k = 1 \quad (**).$$

For the concrete case of the CTMC for memory-less browsing we can compute the expected data rate as follows. Given holding times v_{is} and $v_i, i = 1, \ldots, |L|$ we get $v_i P_i = q_{is,i} P_{is}$ since the states i can only be reached from $state_{is}$ and thus

$$P_i = \frac{v_{is}}{v_i} p_{is,i} P_{is} \quad (* * *)$$

for $i = 1, \ldots, |L|$, using equation $(*)$.

16

Substituting equation $(**)$ for our concrete case with $\sum_{i \in |L|} P_i + P_{is} = 1$ and using equation $(***)$ yields

$$P_{is} = \frac{1}{1 + \sum_{i=1,\ldots,|L|} \frac{v_{is}}{v_i} p_{is,i}}$$

from which the other values P_i can be immediately derived.

The expected value $E(i)$ of resources required by state i within the long-run analysis is determined then by

$$E(i) = P_i * res(i)$$

where $res(i)$ is the amount of resources consumed in state i.

The expected amount of resources required by a browsing session is then

$$E_{session} = \sum_{i \in I} E(i).$$

Since idle states do not consume resources, the expected resource demand is then computed as

$$E_{session} = \sum_{i=1,\ldots,|L|} P_i * rate(scene_i).$$

This derivation shows that for the CTMC for memory-less browsing we can derive the equilibrium distribution and, thus, the expected resource demand in linear time cost in the size of the result list.

For our running example, we obtain by means of using equation $(***)$ and the v_i- and p_i-values from section 3.3 the equilibrium probabilities

$$P_{is} = 0.31, P_1 = 0.14, P_2 = 0.32, P_3 = 0.21, P_4 = 0.01, P_5 = 0.005.$$

Note, though the first video has higher relevance the probability that the system is in the state of presenting the second or third video is higher. This is due to the fact that those videos have substantially longer holding times. The expected resource demand for this session is then $E_{session} = 1.35 Mb/s$.

4.3 Admission Control of Pending Clients

The multimedia database systems limit the number of active clients that are allowed to simultaneously perform a browsing session for inspecting the hit list. Thus when a client issues a query, the results will only be presented if sufficient resources are available. For determining whether sufficient resources are available the prediction models introduced in the previous section are employed. A client that has been admitted will be served for the complete browsing session, with high probability in the required quality.

Let us assume the system has already admitted clients $c_1, \ldots, c_k$ and a new client c_p requests admission. Then the admission control mechanism computes the expected resource demand of the running clients $E(c_j), j = 1, \ldots, c_k$ and the expected resource demand $E(c_p)$ of the new client. Then the admission criterion is

$$E(c_p) + \sum_{j=1,\ldots,c_k} E(c_j) < \tau * s_{max},$$

where s_{max} is the amount of maximal available resources and $\tau \in [0,1]$ is a safety margin to allow small deviations from the expected resource usage. The quantity τ determines how close the average load values may approach the maximum server load, and thus how much tolerance is available to compensate for deviations between predicted and real server load. High values of τ represent a permissive admission policy, while low values of τ represent a cautios admission policy. For a large number of possible clients, such a criterion based on an estimation of the average resource usage appears to be appropriate, since deviations from the average values of single clients can be expected to compensate for statistical reasons. For a small number of clients, other admission criteria based, for example, on maximum expected resource usage or maximum expected deviation, can be considered in addition.

The actual resource usage of a client can be determined a posteriori by analyzing its requests to the system. This technique has been used in [HA98] to devise an alternative admission control mechanism, based on the lookback to past system behavior. It may occur that the predicted resource usage of a client and the actual resource usage systematically deviate from each other. In such a case, it is quite clear, that one can use the information on the actual behavior to systematically correct future predictions. A detailed discussion of this approach is, however, beyond the scope of this paper.

For the concrete realization of the admission control mechanism a number of further issues need to be resolved, like the definition of admission points, the treatment of rejected clients, the recomputation of predictions for admitted clients and the reaction to overload situations. Some solutions to that extent have been presented in [HA98], in particular, a complete specification of an admission control algorithm.

5 Related Work

Most approaches to admission control consider the request of single media streams within a server-push architecture. Since the resource requirements of a whole presentation are prespecified by the request the server manages the delivery of the whole media stream and pushes the data continuously to the client. Admission control mechanisms for "pure" server-push architectures assume that during a presentation the data rate consumed by the client will be nearly constant [RVT96]. The available system resources are calculated by stochastic [NMW97], [VGGG94] or deterministic approaches [VGG95], [ORSN96]. Based on the knowledge about the already reserved and freely available resources, it is possible to reject requests in case of server overloads. In the following, we discuss some of the strategies that consider interactive applications.

A priori reservation. To guarantee a given QoS worst-case assumptions about the data rate required can be made. Obviously, the reservation of the worst-case data rate wastes server resources and decreases the number of clients that can be served in parallel. Dey-Sircar et al. [DSSKT94] give stochastical guarantees by means of reserving separate server bandwidth for VCR-interactions. The drawback of their work is that they assume interactions to occur rarely. Most concepts providing stochastical service guarantees assume stochastical retrieval time from storage system which we do not consider. For example, [VGGG94] exploit the variation in access times from disk, but neglect varying user demands.

Re-admission at interaction points.   A straightforward way to use standard admission control policies with interactive applications is to perform admission control for each single media object request that can occur as the result of an interaction as described in Gollapudi and Zhang [GZ96]. One drawback of their approach is that each client request is subject to the admission control. For example, when the first scene of a video is admitted there is no guarantee for the admission of the subsequent scenes of the same presentation. This may lead to unacceptable delay in presentation when too many clients send requests. The main problem of their approach is that the admission is not performed for a client session. Thus, the admission of one continuous media stream of a multimedia presentation does not necessarily guarantee the admission of another continuous media stream that has to be synchronized with the already admitted streams.

Smooth the application data rates.   Some approaches to admission control for interactive applications propose to "smooth" the data rate deviations to achieve a relatively constant workload. Shenoy and Vin [SV95] reduce the high data rate for fast forward and fast rewind of MPEG-videos by encoding the stream in base and enhanced layers. The encoding of the base layer is done by reducing the temporal and spatial resolution. For fast forward, only the base layer is used. Chen, Kandlur, and Yu [CKY94] suggest segment skipping where a segment can be a set of Group of Pictures (GoP) of an MPEG-video. For fast forward or fast rewind, some segments are skipped. Chen, Krishnamurthy, Little, and Venkatesch [CKLV95] change the order of MPEG-frames to a priority sequence. For fast forward and fast rewind, only the most important frames (I- and P-Frames) are pushed to the client. The higher data rate is reduced by quality adaptation on the temporal dimension of other requests by a dynamic resource reservation. Reddy [Red97] reduces the latency of "urgent" requests, but neglects varying bandwidth requirements. The smoothing approach is, however, restricted to relativley simple interactive scenarios where interactions take place within the presentation of one single media stream.

Inspect the past system behavior.   In earlier work, we presented a general admission control mechanism which is applicable for varying resource requirements of highly interactive applications requiring a client-pull architecture [HA98]. It consists of (1) the admission of new clients when server resources are available and (2) the scheduling and adaptation of requests of admitted clients. For the admission of new clients, we inspect the past system behavior. For a large number of parallel sessions, the average client consumption is a good estimate for prediction. Data rate variations are accounted for by introducing a safety margin. Thus, an admitted client is supposed to obtain sufficient resources. If in spite of the admission control resource bottlenecks occur, strategies for rescheduling requests are used to achieve high QoS by means of load balancing. In the worst case quality adaptations are required to enable guaranteed continuous delivery.

Usage of application semantics.   Zhao and Tripathi [ZT98] propose a session-based reservation approach for multimedia applications with varying resource requirements. A multimedia session consists of the presentation of multiple multimedia objects that have to be synchronized in temporal order. The temporal order of the presentation is known at admission time. They propose an "advanced resource reservation" mechanism, i.e., to reserve resources for time intervals in the future. The goal of the approach is to determine a starting point for the presentation for which all required resources (i.e., network and end system) are available. The basic reservation model does not consider user interactions. The extensions they propose for interactions are not

sufficient since they are not economically (i.e., specification of a minimum upper bound) or are not able to bound delays since they suggest re-admission at interaction point as described earlier in this section.

The use of continuous-time Markov chains for modeling the access behavior in a multimedia database system to support the efficient vertical data migration between the tertiary and secondary storage has been devised in [KW97]. This shows that the application of the CTMCs to model resource usage in multimedia databases is not only limited to admission control but is applicable to other aspects of resource management as well.

6 Conclusion

In this paper, we presented an admission control mechanism for highly interactive browsing applications by considering application semantics for the admission of new clients. It is based on the stochastical resource prediction of clients. We assume that the user behavior is related to the relevance values of a conceptual query and specify the user behavior as Continuous Time Markov Chains.

A Java based implementation of the admission algorithm within the IDS based system architecture described in Section 2 is under way. Future work will concentrate on the refinement and evaluation of the approach and on learning models for user profiles. An evaluation of the concept will strongly depend on the availability of sufficient real-world data against which the proposed models can be calibrated. From this data, learning methods which are applicable to the Markov chain models can be exploited to adapt the admission control framework to particular application scenarios. In combination with the retrospective approach to admission control by inspection of past system behavior, a long term vision is a self-adapting admission control framework for multimedia database access.

References

[CKLV95] Huang-Jen Chen, Anand Krishnamurthy, Thomas D. C. Little, and Dinesh Venkatesch. A scalable video-on-demand service for the provision of VCR-like functions. In *Proc. 2nd Intl. Conf. on Multimedia Computing and Systems*, pages 65–72, May 1995.

[CKY94] Ming-Syan Chen, Dilip D. Kandlur, and Philip S. Yu. Support for fully interactive playout in a disk-array-based video server. In *ACM Multimedia*, 1994.

[CT95] Stavros Christodoulakis and Peter Triantafillou. Research and development issues for large-scale multimedia information systems. *ACM Computing Surveys*, December 1995.

[DSSKT94] Jayanta K. Dey-Sircar, James D. Salehi, James F. Kurose, and Don Towsley. Providing VCR capabilities in large-scale video servers. In *ACM Multimedia*, pages 25–32, 1994.

[EHT98] André Everts, Silvia Hollfelder, and Ulrich Thiel. Browsing descriptors and content-based presentation support for videos. In *GI-Workshop Multimedia Sys-*

teme, in conjunction with Informatik'98, (GI-Jahrestagung Informatik zwischen Bild und Sprache), September 1998. To be published.

[GZ96]   Sreenivas Gollapudi and Aidong Zhang. Netmedia: A client-server distributed multimedia environment. In *Proc. of Third Intl. Workshop on Multimedia Database Management Systems*, pages 352–363, August 1996.

[HA98]   Silvia Hollfelder and Karl Aberer. An admission control framework for applications with variable consumption rates in client-pull architectures. In *Proceedings of Multimedia Information Systems (MIS'98)*, Istanbul, September 1998. To be published, also appeared as GMD Technical Report, Sankt Augustin, Nr. 8, April 1998.

[HKR97]   Silvia Hollfelder, Achim Kraiss, and Thomas C. Rakow. A client-controlled adaptation framework for multimedia database systems. In *Proc. of European Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS'97)*, Darmstadt, Germany, Sept 1997.

[HSHA98]   Silvia Hollfelder, Florian Schmidt, Matthias Hemmje, and Karl Aberer. Transparent integration of continuous media support into a multimedia DBMS. In *Proceedings of IADT '98*, Berlin, 1998. Enlarged version published as GMD Technical Report, Nr. 1104, Sankt Augustin, Germany, December. 1997.

[INF97]   INFORMIX Press. *Video Foundation DataBlade Module User's Guide, Version 1.1*, June 1997. Version 1.1.

[KW97]   Achim Kraiss and Gerhard Weikum. Vertical data migration in large near-line document archives based on markov-chain predictions. In *VLDB*, pages 246–255, 1997.

[ME97]   Adrian Müller and André Everts. Interactive image retrieval by means of abductive inference. In *RIAO 97 Conference Proceedings – Computer-Assisted Information Searching on Internet*, pages 450–466, June 1997.

[NMW97]   Guido Nerjes, Peter Muth, and Gerhard Weikum. Stochastic performance guarantees for mixed workloads in a multimedia information system. In *Proc. of the IEEE International Workshop on Research Issues in Data Engineering (RIDE'97)*, Birmingham, UK, April 1997.

[ORSN96]   Banu Özden, Rajeev Rastogi, Avi Silberschatz, and P. S. Narayanan. The Fellini multimedia storage server. In S. M. Chung, editor, *Multimedia Information Storage and Management*. Kluwer Academic Publishers, 1996.

[Red97]   Narasimha Reddy. Improving latency in interactive video server. In *Proc. of SPIE Multimedia Computing and Networking Conference*, pages 108–112, February 1997.

[RVT96]   Siram S. Roa, Harrick M. Vin, and Asis Tarafdar. Comparative evaluation of server-push and client-pull architectures for multimedia servers. In *Proc. of Nossdav 96*, pages 45–48, 1996.

[SN95]   Ralf Steinmetz and Klara Nahrstedt. *Multimedia: Computing, Communications and Applications*. Prentice Hall Series in innovative technology. Prentice Hall, 1995.

[SV95]   Prashant J. Shenoy and Harrick M. Vin. Efficient support for scan operations in video servers. In *Proc. of the Third ACM Conference on Multimedia*, 1995.

[THE98]   Ulrich Thiel, Silvia Hollfelder, and André Everts. Multimedia management and query processing issues in distributed digital libraries: A HERMES perspective. In *Proc. of the Sec. Int. Workshop on Query Processing in Multimedia Information Systems QPMIDS'98, in conjunction with DEXA'98*, August 1998. will be published.

[Tij94]   Henk C. Tijms. *Stochastic Models. An Algorithmic Approach*. Wiley series in probability and mathematical statistics. Wiley, 1994.

[TK96]   Heiko Thimm and Wolfgang Klas. Delta-sets for optimized reactive adaptive playout management. In *Proc. 12th Int. Conf. On Data Engineering (ICDE)*, pages 584–592, 1996.

[TKC⁺97]   Heiko Thimm, Wolfgang Klas, Crispin Cowan, Jonathan Walpole, and Calton Pu. Optimization of adaptive data-flows for competing multimedia presentational database sessions. In *IEEE International Conference on Multimedia Computing and Systems*, pages 584–592, June 1997.

[VGG95]   Harrick M. Vin, Alok Goyal, and Pawan Goyal. Algorithms for designing large-scale multimedia servers. *Computer Communications*, March 1995.

[VGGG94]   Harrick M. Vin, Pawan Goyal, Alok Goyal, and Anshuman Goyal. A statistical admission control algorithm for multimedia servers. In *Proc. of the ACM Multimedia*, pages 33–40, October 1994.

[YYWL95]   M. M. Yeung, B.-L. Yeo, W. Wolf, and B. Liu. Video browsing using clustering and scene transitions on compressed sequences. In *IS &T SPIE Multimedia Computing and Networking*, 1995.

[ZT98]   Wei Zhao and Satish K. Tripathi. A resource reservation scheme for synchronized distributed multimedia sessions. *Multimedia Tools and Applications*, (7):133–146, July 1998.