A Performance Model of Mixed-Workload Multimedia Information Servers[*]

Guido Nerjes, Peter Muth, Gerhard Weikum

University of the Saarland, Department of Computer Science
D-66041 Saarbrucken, Germany
{nerjes, muth, weikum}@cs.uni-sb.de

Abstract

Advanced multimedia applications such as digital libraries or teleteaching exhibit a mixed workload with accesses to both ”continuous” data (e.g., video) and conventional, ”discrete” data (e.g., text/image documents). As the fractions of continuous-data versus discrete-data requests vary over time, we consider a multimedia storage server with both classes of data spread across all disks for dynamic load sharing. This paper develops a stochastic model for predicting the performance of mixed multimedia workloads on a given system configuration. It focuses on analyzing the response-time distribution for discrete-data requests under potential contention with continuous-data requests. We derive an upper bound for the probability that the response time of a discrete-data request exceeds a specified tolerance threshold. Experimental results from detailed simulation studies demonstrate the high accuracy of the analytical model. Thus the model is an appropriate basis for server capacity planning.

1 Introduction

1.1 Problem Statement

Multimedia information servers that manage large volumes of disk-resident video/audio data as well as text and image data have to meet stringent performance requirements. The real-time nature of “continuous” data (*C-data*) like video/audio dictates that a server must meet firm deadlines for each of the data fragments that constitute the data stream between the server and a client throughout the playback of an entire C-data object. For example, a fragment may consist of all video frames that correspond to one second of playback, and fragments that arrive too late at the client (i.e., more than one second after the previous fragment has started its playback) may cause user-noticeable degradations of the playback quality. We refer to such timing problems within C-data streams as *glitches*.

The well-established notion of *quality of service (QoS)* for C-data demands that glitches should either be completely avoided or that the probability or rate with which they occur within a stream is bounded (by a very small number, say 0.001). To guarantee the promised QoS, a multimedia server must exert an admission control to limit the maximum number of concurrently active C-data streams. So, given the number of concurrently active users as well as data and access characteristics of an application, it is absolutely crucial that a server be configured appropriately (i.e., should have the necessary number of disks, amount of memory, etc.).

The server configuration problem is made significantly harder by the fact that many advanced multimedia applications such as digital libraries or teleteaching will exhibit a *mixed workload* with massive access to conventional, “discrete” data (*D-data*) such as text and image documents as well as index-supported searching in addition to the requests for continuous data. Furthermore, with unrestricted 24-hour world-wide access over the Web, such multimedia servers have to cope with a dynamically evolving workload where the fractions of C-data vs. D-data requests vary over time. Thus, for a good cost/performance ratio it is mandatory that a server operates with a shared resource pool rather than statically partitioning all resources (disks, memory, etc.) into two pools, one for each of C- and D-data.

The success of a multimedia information service critically depends on its QoS and responsiveness as perceived by the users. This does, of course, include the response time for accesses to conventional D-data, an issue that has been rather neglected in the literature which has focused almost exclusively on C-data. In this pa-

[*] In: Proc. of 10. ITG/GI-Fachtagung: Messen, Modellieren und Bewerten von Rechen- und Kommunikationssystemen (MMB’99), Trier, September 21-24, 1999

per we study stochastic response-time guarantees for D-data in conjunction with stochastic service-quality guarantees for C-data. For example, an application could demand that the response time of D-data accesses does not exceed 2 seconds with a probability of at least 90 percent (i.e., bounding the tail of the response-time distribution). The ultimate objective of our research is to develop a method for configuring a mixed-workload multimedia information server that meets the application's requirements on both C- and D-data while minimizing the cost of the server. This paper focuses on the mathematical underpinnings of such a configuration tool in that it develops an accurate analytical model for predicting the performance and QoS of a given server configuration.

1.2 Related Work

Fairly detailed analytical models have been developed for multimedia servers, with exclusive focus on C-data requests, however. These models allow predicting, from the data and storage system parameters, the maximum number of concurrent C-data streams that the server can sustain with either no glitch at all [1, 2, 3, 4, 5], with a probabilistic bound on the glitch rate or related delay metrics [6, 7, 8, 9]. In this paper we pursue the latter kind of stochastic model, as opposed to the first type of deterministic worst-case models. Stochastic guarantees for service quality are tolerable for almost all multimedia applications; moreover, glitch situations can often be masked or at least "smoothed" by a carefully controlled, dynamic QoS adaptation (see, e.g., [10]). Furthermore and most importantly, not taking into account the stochastic nature of disk service times (variable seek times, variable transfer rates on multi-zone disks, etc.) and variable-bit-rate-encoded C-data would inevitably lead to overly conservative predictions and thus poor cost/performance ratio of a server.

Among the few papers that have given at least some thought to mixed workloads are [11, 12] and our own prior work [13, 14, 15, 16]. In [11] the impact of the additional D-data requests is taken into account by reserving a fixed fraction of the server's performance capacity for D-data requests. However, this is merely a best-effort approach without any quantitative considerations. In [12] disk scheduling heuristics for mixed workloads are studied by simulation. Both [13] and [14] were preliminary attemps to develop a stochastic model for mixed workloads. The two approaches used relatively crude mathematical models, had to make very simplifying restrictions and thus ended up with fairly inaccurate predictions. The (preliminary) conclusion of our prior work was that one should resort to simulation models for performance predictions. Finally, [15, 16] completely concentrated on simulation experiments in its study of different scheduling policies. In summary, no sufficiently accurate analytical model has been developed so far to be practically useful as the basis for a server configuration tool.

1.3 Contribution and Outline of the Paper

The paper's contribution lies in developing the mathematical underpinnings for a configuration tool for mixed-workload multimedia information servers. To this end, we develop a queueing model that allows us to predict the response time of D-data requests (subsequently abbreviated as *D-requests*) in the presence of C-data requests (*C-requests*). This model complements earlier work of ours [8] on predicting the QoS of a pure C-data workload, The stochastic guarantees for the C-data glitch rate that we derived there and that is summarized in this paper can be carried over to the new mixed-workload model.

In terms of its underlying mathematical techniques, the developed queueing model builds on modeling techniques for special classes of M/G/1 servers that limit the number of served requests in a service period [17]. The derivation of the D-request response time, which is the model's core part, is a fundamentally new challenge, however, because we consider an efficient SCAN policy for the disk arm scheduling whereas the prior work has been restricted to FCFS policies [18]. To the best of our knowledge, the current paper is the first one that considers this problem in the context of multimedia storage management and solves it in a manner that is both accurate and computationally tractable. All derivations are carried out in terms of the Laplace transforms of the underlying probability distributions [19, 20]. Thus we capture entire distributions, not just mean values. In particular, we can make quantitative statements about the tail of the response time distribution (e.g., the 90th percentile), using the Chebyshev inequality or the more accurate but computationally more expensive method of Chernoff bounds derived from the Laplace transforms [21, 20].

The rest of the paper is organized as follows. Section 2 introduces the system architecture of a mixed-workload multimedia data server. Section 3 summarizes the analytical model for predicting the glitch rate of C-data. Section 4 develops the queueing model that allows us to predict the response time of D-requests. Section 5 presents experimental results from a detailed simulation study that demonstrates the accuracy of the presented analytic model for discrete data.

2 Architecture of a Mixed-Workload Server

In this section, we discuss the system architecture for which our approach is geared. Clients submit requests for both C-data and D-data to a server. A C-data object (e.g., a video) is composed of a sequence of *fragments*. The delivery of such an object from the server to the client constitutes a producer-consumer data *stream* throughout the object's playback. D-data objects, on the other hand, are requested as indivisible units (as far as our model is concerned). We assume a fast and reliable network with a performance capacity well above the application's bandwidth requirements, and thus disregard network issues in this paper.

The performance metric we are considering is glitch rate of C-requests for a given number of concurrent streams and the response time of D-requests for a given arrival rate. We want to derive an analytic model that allows us to bound the probability that the response time and the glitch rate exceeds a specified tolerance threshold. Additional metrics such as server memory demand or startup latency are not studied in this paper.

2.1 Data Placement

The following assumptions capture the method of choice that has evolved in the literature (see, e.g., [1, 22, 23, 24, 2, 5, 3]). We consider a server with K disks. Since compression techniques reduce the bandwidth of video/audio objects substantially, we assume that the required bandwidth per C-data stream is always smaller than the transfer rate of a single disk. We allow variable bandwidth both across objects and within a single object, as commonly used compression techniques such as MPEG-2 are based on variable-bit-rate (VBR) encodings. Objects are partitioned into *constant-time-length (CTL) fragments* such that each fragment corresponds to the same fixed playback time T (e.g., one second). Consequently, fragments vary in size even within one object. This scheme has the advantage that the discretization of the C-data streams induces a perfectly regular, periodic access pattern on the server: one fragment for each stream within each time unit.

Fragments are assigned to disks in a round-robin manner, using coarse-grained striping so that each fragment resides entirely on one disk. This scheme maximizes the effective disk bandwidth while balancing the load across all disks. In addition and most importantly, it provides perfect scalability in that the maximum number of concurrently sustainable streams grows linearly with the number of available disks. This holds even if the access frequencies of C-data objects are highly skewed.

D-data objects are allocated on the disks such that the expected I/O load for this data is balanced across all disks. This may involve striping for large objects, with adequately tuned striping units. For this paper, we do not rely on any specific assumptions on the placement and storage layout of D-data objects. As already pointed out in the introduction, we assume that both C- and D-data reside on the same shared disk pool, so that both workload classes can fully exploit the space and performance capacity of all disks. This is extremely beneficial especially when the load fractions of the two classes evolve over time.

2.2 Delivery of and Admission Control for C-Data Streams

The data layout of C-data discussed above is exploited by the disk scheduling, which is periodic and proceeds in *service rounds* of fixed duration T that corresponds to the playback time length of the fragments. In each round the server needs to fetch from each disk those fragments that need to be delivered to the clients by the end of the next round (using server memory for intermediate buffering). After having fetched N fragments from disk i in one round, the server needs to fetch N fragments for the underlying streams from disk $(i + 1) \bmod K$ in the next round, where K is the total number of disks. Not being able to fetch all the necessary fragments by the end of a round is what causes glitches in the affected streams.

This scheme simplifies the admission control for clients that request to start a new C-data stream. We only have to test that the disk that holds the first fragment of the requested object can accomodate the additional load for this fragment in a service round.

Given the periodic shifting of the load pattern, a positive result for this admission test will then imply that the probability of incurring glitches is sufficiently low in all subsequent rounds as well. This stochastic consideration must, however, take into account the variable size of fragments (and further variable parameters such as seek times and rotational delays) based on knowledge of its statistical distribution [8].

Another important property of this admission control scheme is that it needs to consider only the load on a single disk. The regularity of the overall load pattern ensures that the result of the sustainability test carries over to the entire pool of parallel disks. For the same reason, it is sufficient that we consider only a single disk in the performance-prediction model developed in this paper.

2.3 Disk Scheduling

Once the length T of a service round is fixed (typically in the order of 1 second), the actual disk scheduling for C-requests is straightforward. All fragments that need to be fetched from a disk by the end of a round are known at the beginning of the round, so that we can employ a *SCAN algorithm* (also known as “elevator” or “sweep” algorithm) for the disk arm movement, in order to minimize seek times. With this algorithm, requests are sorted according to their seek position on the disk and are served with one sweep of the disk arm.

A number of options are possible, however, when we add D-requests to be served as well. Our earlier work in [15] has discussed these options in a taxonomical manner and identified the most promising schemes. The scheme that we consider in the current paper is a *mixed SCAN* algorithm that includes a number of D-requests

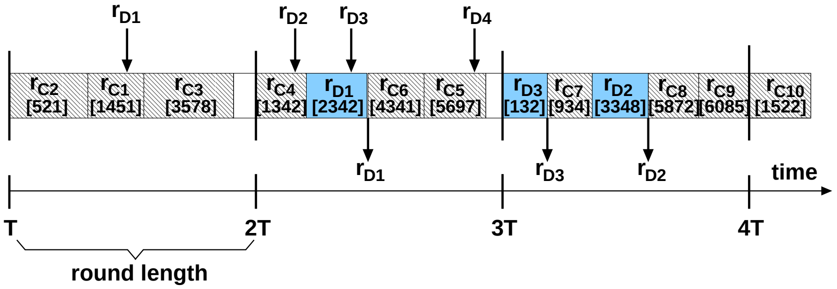

Figure 1: Illustration of the mixed SCAN Disk Scheduling Policy

in the disk sweep of one round. However, the number of D-requests per sweep is limited by a system parameter M such that the service time of all requests (for both C- and D-data) does not exceed the given round length. In addition, to be able to plan the disk schedule on a round by round basis, i.e. to determine the sequence of requests in the disk sweep at the beginning of each round, we employ a *gated* scheduling discipline for D-requests. This means that only those D-requests are eligible for inclusion in the disk sweep that have arrived by the beginning of the round. Furthermore we assume that all requests that are selected in FCFS order at the beginning of a round will be served within this round. Requests that arrive in the middle of a round will be considered for service only in the next disk sweep, i.e. in the next round. This delay is not critical given that the typical round length is one second. The case to be avoided is that too many D-requests are delayed by multiple rounds because of the high overall load (or, equivalently, too few disks for the given load). This case can happen as illustrated by the scenario shown in **Figure 1**. In this figure, the arrivals and departures (i.e., service completions) of D-requests are depicted by arrows. The execution of C- and D-requests is shown in the form of shaded boxes (light for C-requests, dark for D-requests) whose lengths correspond to disk service times. The numbers in parentheses denote the disk cylinder for a request. The time span between the arrival and departure of a D-request is its response time. As shown in the example, request r_{D4} is delayed by more than one round and request r_{D3} is served ahead of request r_{D2} because of the SCAN algorithm.

3 An Analytic Model for the Glitch Rate

This section briefly summarizes our analytic model for the C-data glitch rate, originally developed in [8], putting the model for the D-data response time into perspective.

The goal of analytic modeling is to predict the service quality of a continuous data stream a under given data partitioning, data placement and disk scheduling scheme, given workload parameters such as the number of concurrent streams, the size of fragments, and so on. Such a prediction is of crucial importance for two reasons:

- *Admission control:*
  to calculate the workload that can be sustained for a given configuration and service quality demands, so as to determine how restrictive the admission control needs to be
- *System configuration:*
  to calculate the number of disk resources needed to sustain a given workload under specified service quality demands

Service quality can be measured as the number or rate of video/audio frames that are not delivered to the client according to the video/audio object's timeline for smooth playback, often casually referred to as "hiccups". As far as the server is concerned, these kinds of errors are produced when fragments are not retrieved and delivered just in time, i.e., during the time window of a scheduling round. The goal then is to either guarantee that no glitches occur or that a limited numbers of glitches occur with a very small, more or less negligible probability. Once these service quality demands are specified, analytical modeling then serves to derive the maximum acceptable number of concurrent streams.

The developed stochastic model gives service quality guarantees for a continuous data server with multi-zone disks and VBR encoded continuous data objects. Here workload *and* disk characteristics are modeled in a stochastic way using Laplace transforms and Chernoff bounds for the tail of a probability distribution [21, 20]. This approach goes significantly beyond the much simpler models published so far. Our approach captures all details of the disk system in a realistic manner and provides very tight stochastic bounds, thus allowing us to provide a specified service quality guarantee with much less disk resources than the previously published models. The presentation here gives a general outline. For a detailed derivation of formulas see the full paper in [8].

The overall goal is to bound the probability that the retrieval of a continuous data fragment does not meet its delivery deadline imposed by the real time playback constraints. This probability allows predicting and, consequently, bounding the rate at which the presentation of a continuous data object may suffer “glitches”. Conversely, with a specified tolerance threshold glitch-probability, it is possible to determine the minimum amount of resources (i.e., number of disks) that are necessary to guarantee this quality of service specification.

3.1 Data Model

It is assumed that the data has been compressed using variable bit-rate techniques such as MPEG-2. Due to the normalization of all fragments to the same time length (i.e., the CTL partitioning) the fragment sizes vary. It is shown in [25, 26] that the distribution of fragments sizes can be statistically described by a lognormal or Gamma distribution [19]. In the following analysis, the probability density f_{size_C} of the fragment size distribution is given by

$$f_{size_C} = \frac{\alpha(\alpha x)^{\beta-1} e^{-\alpha x}}{\Gamma(\beta)} \qquad (1)$$

where Γ denotes the Gamma function. The parameters α and β depend on the characteristics of the continuous data and are chosen based on the mean value of fragment sizes $E[S]$ and the variance $Var[S]$.

$$\alpha = \frac{E[S]}{Var[S]} \qquad \beta = \frac{(E[S])^2}{Var[S]} \qquad (2)$$

3.2 Disk Model

Multi-zone disks group adjacent tracks into a number of zones. Each zone has a fixed number of sectors that are allocated in a track, but this number differs between different zones. Inner zones, i.e. zones that are located near the center of the disk, support less sectors per track than outer zones due to the constant aerial recording density. As the angular velocity is kept constant, outer zones provide a higher transfer rate than inner zones [27, 28]. Since typical high performance disks have a space capacity and transfer rate ratio between outer and inner tracks of a factor of two, this is clearly an important performance factor.

We assume that data is uniformly distributed over all sectors of the disk. Therefore the variable track capacity induces a skewed access frequency distribution for the tracks with a higher probability of outer tracks. Given a multi-zone disk with c_z zones, a track capacity of c_{min} for the innermost track and c_{max} for the outermost track, and the time c_{rot} for a single revolution of the disk, the probability density of the transfer rate is given by f_{rate}.

$$f_{rate}(r) = \frac{2rc_z - 2r + c_{max}/c_{rot} - c_{min}/c_{rot}}{(c_{min} + c_{max})c_z(c_{min} - c_{max})/c_{rot}^2} \qquad (3)$$

3.3 Bounding the Total Service Time Per Round

The total service time T_N for the retrieval of N fragments from a single disk in one round is the sum of the seek time $T_{seek,i}$, the rotational delay $T_{rot,i}$, and the transfer time $T_{trans,i}$ for each fragment i. The sum of all seek times in a round can be approximated for the scan algorithm using a tight upper-bound constant T_{seek} with $\sum_{i=1}^{N} T_{seek,i} \leq T_{seek}$ [29] based on the concavity of the seek time as a function of the seek distance [27]. Thus we obtain:

$$T_N = T_{seek} + \sum_{i=1}^{N} T_{rot,i} + \sum_{i=1}^{N} T_{trans,i} \qquad (4)$$

All N random variables $T_{rot,i}$ are independently and identically distributed with a uniform distribution between 0 and the time for one disk revolution. Similarly, the random variables $T_{trans,i}$ are independently identically distributed. This distribution depends on the distribution of the fragment size in equation (1) and the disk transfer rate in equation (3) and can be computed using a convolution-like integral.

The next step is to bound the tail probability p_{late} of the random variable T_N. It is possible to apply Chernoff’s theorem [21, 20] to calculate a bound b_{late} since the Laplace transform T_N^* of T_N can be calculated using the Laplace transforms of its summands.

$$\begin{aligned} p_{late}(N,t) &= P[T_N \geq t] \\ &\leq \inf_{\theta \geq 0}\{e^{-\theta t} T_N^*(-\theta)\} = b_{late}(N,t) \qquad (5) \end{aligned}$$

The right hand side of inequation (5) can be efficiently evaluated using standard numerical methods. The inequation states that the total service time for N requests exceeds the time t, e.g. the duration of a round of say 1 second, with a certain probability that is definitely less than or equal to b_{late}, e.g., 1 or 0.1 percent. Conversely, once we specify the tolerated lateness probability, i.e., a threshold value for b_{late}, and we consider a round length of T for t, then we can derive the maximum number of concurrent streams per disk, N, that we can sustain without violating the specified stochastic guarantee.

3.4 Bounding the Glitch Rate Per Stream

So far we have considered only the phenomenon of one or more glitches occurring within one round. Our final goal, however, is to guarantee that a single stream that runs for a certain of rounds does not suffer more than a specified small fraction of glitches. The first step towards this goal is to calculate the probability p_{glitch} and its corresponding bound b_{glitch} for the event that a certain stream suffers a glitch during a single round. We assume that the streams that are affected by glitches are

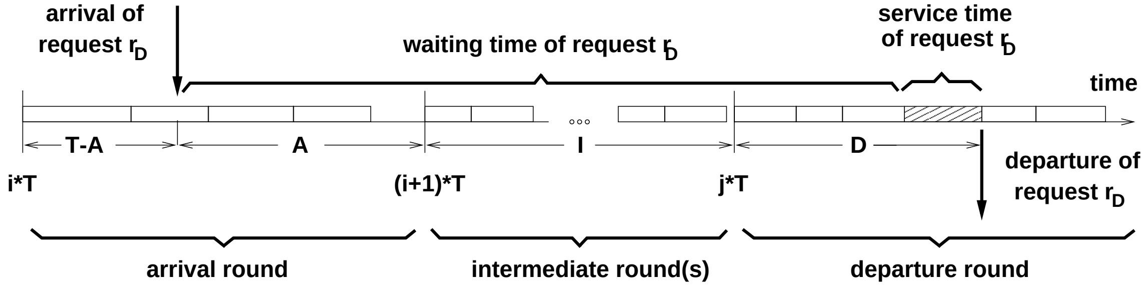

Figure 2: Components of the D-Request Response Time

"selected" independently among the rounds (i.e., there is no artificial correlation across rounds in this respect).

$$\begin{aligned} p_{glitch}(N,T) &= \frac{1}{N}\sum_{k=1}^{N} p_{late}(k,T) \\ &\leq \frac{1}{N}\sum_{k=1}^{N} b_{late}(k,T) = b_{glitch}(N,T) \quad (6) \end{aligned}$$

The probability p_{error} that a stream suffers g glitches during its lifetime of C rounds is characterized by a binomial distribution. Again the tail probability can be bounded by another variant of Chernoff's theorem, derived in [30]

$$\begin{aligned} p_{error}(N,T,M,g) &\leq \\ &\left(\frac{M b_{glitch}(N,T)}{g}\right)^{g} \\ &* \left(\frac{M - M b_{glitch}(N,T)}{M-g}\right)^{M-g} \quad (7) \end{aligned}$$

Inequation (7) expresses the final stochastic guarantee that can be given by the developed model. We can assure that the probability that a stream with a duration of C rounds suffers more that g glitches, given a total load of N concurrent streams and a round length of T, will be less than the right hand side of inequation (7). Conversely, for a specified error probability threshold $b_{error}(N,T,C,g)$, and fixed values of T, C, and g, it is possible to derive the maximum feasible number of concurrent streams, N_{max}, characterized by:

$$N_{max} = \max(N : p_{error}(N,T,C,g) \leq b_{error}) \quad (8)$$

For the validation of the model, detailed simulations were carried out, and the simulation results were compared to the values derived from formula (7) and (8). For details see [8]. It turned out that the derived analytic bound is conservative with only a small deviation compared to the simulation results. In contrast, deterministic worst-case models would significantly overestimate the total service time per round and would thus end up substantially underloading the disk resources. As shown in [8] for the same scenario a deterministic model looses more than a factor of two in terms of throughput compared to a stochastic model.

4 Analytic Model for the Response Time

In this section we develop a stochastic model for the response time of D-requests with the method of supplementary variables [31, 17]. From the viewpoint of a D-request, we can distinguish three different types of service rounds: a D-request arrives in its *arrival round*, and may have to wait several *intermediate rounds* before it is finally served in its *departure round*. We assume a gated service policy, i.e., the set of D-requests to be served in a round is determined at the beginning of the round.

Given the three types of rounds, the response time of a D-request r_D is given by the sum of the times that r_D spends in each type of round as follows (see **Figure 2**):

1. After arrival, r_D has to wait for its arrival round to finish. This time is denoted by the random variable A.

2. Depending on the server load, r_D has to wait zero or more complete intermediate rounds, in which C-requests and previously queued D-requests are served. We model this time by the random variable I. Future arriving D-requests do not influence the number of intermediate rounds due to the FCFS gated selection strategy.

3. In the departure round of r_D, a number of C-requests and D-requests are included in a SCAN schedule (i.e., a disk sweep). The total service time of the requests that precede r_D in the SCAN order plus the service time of r_D is denoted by the random variable D.

The random variables I and D depend on the D-requests which are in the queue at the arrival point of r_D, and on the C-requests to be served in r_D's intermediate rounds and in its departure round. As discussed in Section 2, we use a number limit for the C-requests and D-requests to be served in a round. Let M be a system parameter that denotes the maximum number of D-requests that can be served per round. For the tractability of the following analysis, we assume that the waiting queue for D-requests has a finite capacity with L_{max} places. Let P_B denote the blocking probability, i.e., the probability that a newly arriving request finds the queue already full and is rejected. We denote the length of the D-request queue (at arbitrary time points) by the random variable L.

In the following we denote the probability density function $f_X(x)$ of a random variable X as $P[X = x]$. This is a slight misuse of notation for continuous random variables, but it makes formulas more readable and avoids lengthy subscripts, especially when we consider joint distributions and conditional distributions. A complete list of all variables used in our analysis can be found in Appendix.

For the response time R of D-requests that are not blocked we obtain:

$$\begin{aligned}
&P\left[R = r \,\middle|\, \begin{array}{c}\text{request not}\\ \text{blocked}\end{array}\right]\\
&= P\left[A + I + D = r \,\middle|\, \begin{array}{c}\text{request not}\\ \text{blocked}\end{array}\right]\\
&= \frac{1}{1-P_B}\, P\left[A + I + D = r \wedge \begin{array}{c}\text{request not}\\ \text{blocked}\end{array}\right]\\
&= \frac{1}{1-P_B} \sum_{l=0}^{L_{max}-1} \int_0^T P\left[I + D = r - t\right.\\
&\qquad\qquad \left.\wedge A = t \wedge L = l\right] dt\\
&= \frac{1}{1-P_B} \sum_{l=0}^{L_{max}-1} \int_0^T P\left[I + D = r - t \mid\right.\\
&\qquad \left. A = t \wedge L = l\right] * P\left[A = t \wedge L = l\right] dt \qquad (9)
\end{aligned}$$

In what follows we conceptually "trace" the processing stages of a D-request r_D. Let I_l, D_l, and A_l denote the random variables corresponding to I, D, A, respectively, under the condition that $L = l$ holds for the queue length at the arrival point of the request. I_l and D_l are independent of r_D's arrival time within its arrival round. Hence we obtain:

$$\begin{aligned}
&P\left[R = r| \begin{array}{c}\text{request not}\\ \text{blocked}\end{array}\right] && (10)\\
&= \frac{1}{1-P_B} \sum_{l=0}^{L_{max}-1} \int_0^T P\left[I_l + D_l = r - t\right]\\
&\qquad\qquad * P\left[A_l = t\right] dt && (11)
\end{aligned}$$

For a fixed l the integral in equation 11 corresponds to the convolution of the independent random variables A_l, I_l and D_l. For the Laplace transform of the response time of D-requests, we therefore obtain:

$$R^*(s) = \frac{1}{1-P_B} \sum_{l=0}^{L_{max}-1} A_l^*(s) * I_l^*(s) * D_l^*(s) \quad (12)$$

Due to the definition the Laplace transform of every probability distribution must yield 1 for $s = 0$, we can derive from equation 12 the blocking probability for a request to be rejected because of a full queue:

$$\begin{aligned}
&\lim_{s\to 0} R^*(s) = 1 \Leftrightarrow\\
&P_B = 1 - \lim_{s\to 0} \sum_{l=0}^{L_{max}-1} A_l^*(s) * I_l^*(s) * D_l^*(s) \quad (13)
\end{aligned}$$

The Laplace transform of I_l is given by:

$$\begin{aligned}
I_l^*(s) &= \int_0^\infty e^{-sx} P\left[I_l = x\right] dx\\
&= \int_0^\infty e^{-sx} P\left[I = x \mid A = t \wedge L = l\right] dx\\
&= e^{-sT*\lfloor \frac{l}{M} \rfloor} && (14)
\end{aligned}$$

The distribution of D_l depends on the service strategy in the departure round. The Laplace transform of D_l for the mixed SCAN strategy that we employ will be derived in the following subsections.

The Laplace transform A_l^* of A_l is defined as

$$\begin{aligned}
A_l^*(s) &= \int_0^T e^{-st} P[A_l = t] dt\\
&= \int_0^T e^{-st} P\left[A = t \wedge L = l\right] dt && (15)
\end{aligned}$$

For deriving A_l^*, we need the distribution of the queue length at the beginning of a round, as derived in the next subsection.

4.1 Queue Length Distribution at the Beginning of a Round

To derive the distribution of the queue length at the beginning of a round, we consider an embedded Markov chain with the starting points of service rounds as embedding points. At each of the time points, $t \in \{t_0, t_1, \dots\}$, marking the beginning of a new round, the system state is given by the number of D-requests in the waiting queue, denoted as L_{t_i}. The steady state probabilities of this system are defined as follows:

$$p_n = \lim_{i\to\infty} P[L_{t_i} = n] \quad (16)$$

We assume that at the beginning of each round, up to M requests are removed from the queue for service within the round. We disregard in our analysis the case that the total service demand of these M requests exceeds the round length. The value of M can be set such that that case will happen only with extremely low probability and is thus indeed negligible. Incoming D-requests that arrive during a round are put into the waiting queue and must wait at least for the beginning of the next round.

Let a_k be the probability that k requests arrive during a round of length T. Assuming that the arrivals follow a Poisson process with arrival rate λ this probability is given by

$$a_k = P\left[\begin{array}{c}k \text{ arrivals during a}\\ \text{round of length } T\end{array}\right] = \frac{(\lambda T)^k}{k!} * e^{-\lambda T} \qquad (17)$$

As we already stated the waiting queue is finite with L_{max} places. The steady state probabilities must satisfy the following three Chapman-Kolmogorov equations (also known as flow balance equations) for the Markov chain at hand:

$$\begin{aligned}
&\text{for } n < L_{max}:\\
&p_n = a_n * \sum_{i=0}^{M} p_i + \sum_{i=M+1}^{\min(M+n,L_{max})} p_i * a_{n-i+M}\\
&\text{for } n = L_{max}:\\
&p_n = \sum_{i=0}^{M} p_i * \sum_{i=L_{max}}^{\infty} a_i + \sum_{i=M+1}^{L_{max}} p_i * \sum_{j=L_{max}-i}^{\infty} a_{j+M}\\
&\text{and}\\
&\sum_{i=0}^{L_{max}} p_i = 1
\end{aligned}$$

The linear equations above can be used to solve for the unknown p_n. Since they form a set of linear dependent equations, the normalization constraint must be used to replace one of the other equations to obtain a solution [19, 20].

4.2 Analysis of the Arrival Round

Based on the steady state probabilities p_i, we can now derive A_l^*. The number L of requests in the queue at the arrival of a new request r_D is determined by the queue length at the beginning of the arrival round plus the number of requests that arrive in the arrival round before r_D. We denote the number of arrivals in a time period of length t by $O(t)$, and thus the number of arrivals within the arrival round before the arrival of r_D happens is $O(T-A)$. We then obtain $P[A_l = t] = P[A = t \wedge L = l]$ under the assumption that all elected requests at the beginning of a round will be served within this round by considering the two cases:

$$\begin{aligned}
&\text{for } l < L_{max}:\\
&P[A = t \wedge L = l] =\\
&\sum_{j=0}^{M} p_j \; P[O(T-A) = l \wedge A = t] +\\
&\sum_{j=M+1}^{\min(M+l,L_{max})} p_j \; P[O(T-A) = l - j + M\\
&\qquad \wedge A = t]
\end{aligned} \qquad (18)$$

$$\begin{aligned}
&\text{for } l = L_{max}:\\
&P[A = t \wedge L = l] =\\
&\sum_{j=0}^{M} p_j (1 - \sum_{m=0}^{L_{max}-1} P[O(T-A) = m \wedge A = t]) +\\
&\sum_{j=M+1}^{L_{max}} p_j (1 -\\
&\sum_{m=0}^{L_{max}-j+M-1} P[O(T-A) = m \wedge A = t])
\end{aligned} \qquad (19)$$

$P[O(T-A) = l \wedge A = t]$ in equation 18 and 19 can be calculated based on the Poisson arrival of D-requests:

$$\begin{aligned}
&P[O(T-A) = l \wedge A = t]\\
&\quad = \; P[O(T-A) = l \mid A = t] * P[A = t]\\
&\quad = \; \frac{\lambda^l \, (T-t)^l}{l!} * e^{-\lambda(T-t)} * \frac{1}{T}
\end{aligned} \qquad (20)$$

Its Laplace transform is derived as follows and solved in [17]. We obtain

$$\begin{aligned}
&\int_0^T e^{-st} P[O(T-A) = l \wedge A = t] \, dt\\
&\quad = \; \int_0^T e^{-st} \frac{\lambda^l \, (T-t)^l}{l!} * e^{-\lambda(T-t)} * \frac{1}{T} \, dt\\
&\quad = \; \frac{1}{(\lambda - s)T} \left(e^{-sT} \left(\frac{\lambda}{\lambda - s}\right)^l - \right.\\
&\qquad\qquad \left. \sum_{m=0}^{l} a_m \left(\frac{\lambda}{\lambda - s}\right)^{l-m} \right)
\end{aligned} \qquad (21)$$

The Laplace transform A_l^* of A_l in equation 15 can now be derived in a straightforward manner by substituting the result of equation 21 into the Laplace transform of $P[A = t \wedge L = l]$ according to equations 18 to 19, and performing some algebraic simplifications.

4.3 Analysis of the Departure Round

In this section, we derive the Laplace transform of the time D_l that a request spends in its departure round under the condition that the queue length L at the time of its arrival was l. We consider a mixed SCAN disk scheduling policy that performs a single sweep over the disk serving both D-requests and C-requests in the order of their cylinder numbers.

Let B_D^* denote the Laplace transform of the service time B for a single D-request r_D, including rotational delay and transfer time, B_C^* is the analogous Laplace transform of the service time for a C-request.

To obtain the seek time distribution for the SCAN algorithm we have to consider two cases: the initial seek to the cylinder of the first request and the subsequent seeks. Let $B_{scan1}^*(n,s)$ and $B_{scanX}^*(n,s)$ denote the Laplace transforms of these random variables. The parameter n indicates that the seek time between two successive requests in a SCAN depends on the number of requests, n, in the SCAN. We obtain the following term for D_l^* (LT stands for Laplace transform):

$$D_l^*(s) = \sum_{d_1=1}^{M} q_{d_1} \sum_{p=1}^{d_1+N} \frac{1}{d_1+N} * \sum_{d_2=\max(0,p-N-1)}^{min(p-1,d_1-1)} pred(N,d_1,p,d_2) * \left(\begin{array}{c}\text{LT of}\\ \text{service time}\end{array}\right) \tag{22}$$

$$\left(\begin{array}{c}\text{LT of}\\ \text{service time}\end{array}\right) = B_{scan1}^*(d_1+N,s) * [B_{scanX}^*(d_1+N,s)]^{p-1} * [B_D^*(s)]^{d_2+1} * [B_C^*(s)]^{p-1-d_2} \tag{23}$$

In the outermost sum we iterate over the number of D-requests that are served before the request in focus in the departure round. q_i denotes the probability that in the departure round of r_D, the number of D-requests to be served in that round is i. q_i is obtained from the steady state probabilities p_n by weighting them with the number of D-requests served in the round:

$$i = 1 \ldots M-1: \quad q_i = \frac{i * p_i}{\sum_{n=1}^{M-1} n * p_n + M * (1 - \sum_{n=1}^{M-1} p_n)}$$

otherwise:

$$q_M = \frac{M * (1 - \sum_{n=1}^{M-1} p_n)}{\sum_{n=1}^{M-1} n * p_n + M * (1 - \sum_{n=1}^{M-1} p_n)}$$

Note that the case of M D-requests served in a round needs special treatment, as M D-requests are served if the queue length at the beginning of the round is at least M.

The sum in the middle of Equation 22 iterates over the position a D-request can take. There are $d_1 + N$ possible positions. The innermost sum iterates over the number of possible D-request and C-request predecessors (in the SCAN) at a given position and a given number of D-requests in the departure round. Assuming that there are N C-requests and d_1 D-request to be served in the departure round of a D-request r_D, r_D can have up to $d_1 + N - 1$ predecessors that are served before r_D depending on its position in the SCAN. By numbering these positions from 1 to $d_1 + N$, we can calculate the probability $pred(N,d_1,p,d_2)$ that at a distinct position p the D-request has d_2 D-requests and $(p-1-d_2)$ C-requests as predecessors. The probability $pred(N,d_1,p,d_2)$ can be computed combinatorially:

$$pred(N,d_1,p,d_2) = \frac{\binom{d_1}{d_2+1}\binom{N}{p-1-d_2}(p-1)!(d_1+N-p)!(d_2+1)}{(d_1+N-1)!d_1} \tag{24}$$

The numerator of equation 24 multiplies the number of possibilities to select $d_2 + 1$ D-requests out of d_1 with the number of possibilities to select $p-1-d_2$ C-requests out of N. These requests are placed on the SCAN positions $1 \ldots p$. There are $(p-1)!$ permutations possible preceding position p and $(d_1 + N - p)!$ following position p. Furthermore there are $(d_2 + 1)$ possibilities to select a D-request for position p. The denominator of equation 24 calculates the total number of possibilities to place d_1 D-requests and N C-requests on $d_1 + N$ positions under the condition that a D-request can be found at position p.

Finally, in order to completely determine $D_l^*(s)$ for the mixed SCAN service policy, the last two steps are to compute the service time distributions for D-requests and C-requests as well as the underlying seek time distributions. This is done in the next two subsections.

4.4 Service Time

The service time for D- and C-requests consists of three components, the rotational delay, the transfer time and the seek time. In this section we derive Laplace transforms for the transfer and the rotational delay, the seek time is considered separately in the next section.

We assume the rotational delay to be uniformly distributed between 0 and c_{rot}. The Laplace transform B_{rot}^* of this distribution is given by [19, 20]:

$$B_{rot}(t) = \frac{1}{c_{rot}} \Rightarrow B_{rot}^*(s) = \int_0^{c_{rot}} e^{-st} B_{rot}(t)\, dt = \frac{1 - e^{-s*c_{rot}}}{s * c_{rot}} \tag{25}$$

Let f_{size_D} be the density of the request size distribution for D-requests. Assuming a constant transfer rate c_{trans}, the Laplace transform of the transfer time distribution is given by

$$B^*_{read_D}(s) = \int_0^\infty e^{-st} f_{size_D}(c_{trans} * t) * c_{trans}\, dt \tag{26}$$

The integral can be symbolically solved for typical request size distributions such as Gamma distributions [8]. Multiplying the Laplace transform of these two components yields the Laplace transform of the service time excluding the seek time

$$B^*_D(s) = B^*_{rot}(s) * B^*_{read_D}(s) \tag{27}$$

B^*_C found in equation 23 can be calculated in the same way using the C-request specific request size distribution f_{size_C}.

4.5 Seek Time

In a SCAN, the seek distance between two successive requests, depends on the number of requests, n, that are served in the disk sweep. This is expressed in the following by the subscript n. We assume that the disk has c_{cyl} cylinders numbered from 1 to c_{cyl}. First we consider the seek time distribution for the seek between cylinder 1 of the disk and the cylinder where the data of the first request in the SCAN resides. The Laplace transform $B^*_{scan1}(n,s)$ of this distribution can be computed using the seek distance distribution $f_{scan1}(n,d)$ and applying the substitution rule for integrals:

$$B^*_{scan1}(n,s) = \int\limits_{seek(1)}^{seek(c_{cyl}-1)} e^{-st} f_{scan1}(n, seek^{-1}(t)) * (seek^{-1})'(t) \;\; dt \tag{28}$$

The seek time function $seek(x)$ used in equation 28 computes the seek time for seeking over x cylinders. Seek time functions for modern disks can be found in [27] (see also Section 5). Its inverse, that is, the function that computes the seek distance for a given seek time, is denoted by $seek^{-1}(t)$. The first derivative of the inverse is given by $(seek^{-1})'(t)$. The seek distance distribution $f_{scan1}(n,d)$ can be computed combinatorially as follows.

Assume that there are n requests $(r_1, r_2, \ldots, r_n)$ ordered by increasing cylinder numbers $(cyl(r_1) < cyl(r_2) < \ldots, < cyl(r_n))$. If the seek distance between cylinder 1 and the cylinder of request r_1 is d then there are $\binom{c_{cyl}-d-1}{n-1}$ possibilities to place the remaining $n-1$ requests on the disk cylinders, provided that all requests are placed on different cylinders. This assumption does not cause noticeable inaccuracies as the probability that two requests hit the same cylinder is negligibly small considering typical values for c_{cyl} [27] (see also Section 5). For each of these possibilities there are $n!$ possible configurations for an ordered tuple consisting of the cylinder numbers of the requests $(cyl(r_{i_1}), \ldots, cyl(r_{i_n}))$. The total number of possibilities to place n requests on c_{cyl} cylinders without having two requests on the same cylinder is given by: $c_{cyl} * (c_{cyl}-1) * (c_{cyl}-2) * \ldots * (c_{cyl}-n+1) = \frac{c_{cyl}!}{(c_{cyl}-n)!}$. This corresponds to the number of ordered n-tuples of different elements from a set of c_{cyl} elements. So the seek distance density for the first seek in the SCAN algorithm starting from cylinder 1 is given by

$$\begin{aligned} & f_{seek1}(n,d) \\ &= P\left[\begin{array}{c|c} \text{seek distance of 1st} & n \text{ requests} \\ \text{seek is } d \text{ cylinders} & \text{in SCAN} \end{array}\right] \\ &= \binom{c_{cyl}-d-1}{n-1} * n! \Big/ \frac{c_{cyl}!}{(c_{cyl}-n)!} \end{aligned} \tag{29}$$

The seek distance distribution f_{scanX} between two successive requests in the SCAN and thus the Laplace transform B^*_{scanX} of the seek time between two requests can be derived in a similar way. If the distance between r_1 and r_2 is d cylinders, then there are $c_{cyl} - (n+d) + 2$ possibilities to place request r_1 and request r_2 on c_{cyl} cylinders, provided that all requests are placed on different cylinders.

The remaining $(n-2)$ requests can be placed on all remaining cylinders with cylinder numbers greater than the cylinder number of request r_2. There are $\binom{c_{cyl}-d-i}{n-2}$ possibilities to assign these $(n-2)$ requests to cylinders if request r_1 resides on cylinder i. For each of these possibilities there are $n!$ possibilities to create ordered n-tuples consisting of the cylinder numbers of the requests $(cyl(r_{i_1}), \ldots, cyl(r_{i_n}))$. Now the total number of possibilities to place n requests on c_{cyl} cylinders under the condition that the distance between the request r_1 and r_2 is d is the sum of all possibilities described above: $\sum_{i=1}^{c_{cyl}-(n+d)+2} \binom{c_{cyl}-d-i}{n-2} * n!$. Since the probability of a seek distance of d between two successive requests is the same for all pairs of successive requests, we can use the above considerations for request r_1 and request r_2 to derive the seek distance density

$$\begin{aligned} & f_{scanX}(n,d) \\ &= P\left[\begin{array}{c|c} \text{seek distance between two} & n \text{ requests} \\ \text{requests is } d \text{ cylinders} & \text{in SCAN} \end{array}\right] \\ &= \sum_{i=1}^{c_{cyl}-(n+d)+2} \binom{c_{cyl}-d-i}{n-2} * n! \Big/ \frac{c_{cyl}!}{(c_{cyl}-n)!} \end{aligned}$$

For B^*_{scanX} we obtain:

$$B^*_{scanX}(n,s) = \int\limits_{seek(1)}^{seek(c_{cyl}-1)} e^{-st} f_{scanX}(n, seek^{-1}(t)) * (seek^{-1})'(t) \;\; dt \tag{30}$$

number of cylinders	c_{cyl}	6720
transfer rate	c_{trans}	8.79 [MBytes/s]
revolution time	c_{rot}	8.34 [ms]
seek time function $seek(d) = \begin{cases} 1.867E{-}3 + 1.315E{-}4\sqrt{d}\ [s] : d < 1344 \\ 3.863E{-}3 + 2.1E{-}6d\ [s] : d \geq 1344 \end{cases}$		

Table 1: Disk Parameters

$D - requests$
$f_{size_D}(x) = \frac{1}{\sigma\sqrt{2*\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ $\mu = 200000 \;\; \sigma = 50000$
$C - requests$
$f_{size_C}(x) = \frac{\alpha(\alpha x)^{\beta-1}}{\Gamma(\beta)} e^{-\alpha x}$ $\alpha = \frac{600000}{200000^2} \;\; \beta = \left(\frac{600000}{200000}\right)^2$

Table 2: Request Size Distributions

In the derivation of $B^*_{scanX}(n,s)$ and $B^*_{scan1}(n,s)$ we ignored the fact that there could be seeks within a request because the required time for intra request seeks is very small.

4.6 Putting Everything Together

In the previous subsections, we have derived all components of the Laplace transform R^* of the response time distribution of D-requests. As the Laplace transform of a random variable contains all information about its distribution, we have thus achieved our goal with regard to predicting the performance of D-requests in a mixed-workload server.

However, from the Laplace transform of a random variable one can not directly obtain the probability that a response time of a D-request exceeds a specified tolerance threshold. In general, this requires inverting the Laplace transform. Unfortunately, complex Laplace transforms can not be easily inverted. This is also the case for the results derived in this paper. However, one can easily derive specific results like the mean value and the second moment (and thus the variance) of a random variable X from its Laplace transform X^* as follows [19, 20]:

$$E[X] = -\frac{dX^*}{ds}(0) \quad E[X^2] = \frac{d^2X^*}{ds^2}(0) \qquad (31)$$

In order to bound the tail of the response-time distribution (e.g., its 90th or 99th percentile), Chernoff's theorem can be used. Namely, the following inequation holds [21, 20]:

$$P[R \geq r] \leq \inf_{\theta \geq 0} \{ \, e^{-\theta t} R^*(-\theta) \, \} := \inf_{\theta \geq 0} \{ \, h(\theta) \, \} \qquad (32)$$

For the given form of h, differentiating h and solving $h' = 0$ for θ yields the optimum value of θ to obtain the tightest possible bound in the Chernoff inequation.

Given this bound on the tail of the response time distribution for D-requests, we are ready to configure multimedia servers with mixed workloads such that exceeding a certain tolerance threshold for the response time of D-requests occurs only with a specified, very small probability.

5 Experimental Validation

5.1 Simulation Testbed

Our experimental testbed consists of a synthetic load generator, a storage server with a single disk that employs the mixed SCAN scheduling algorithm of the queueing model, and a detailed simulation of the disk considering seek time, rotational and transfer delays. The disk parameters are given in **Table 1**. The distributions of the request sizes for C- and D-requests are given in **Table 2**. These values reflect typical MPEG-2 data with a mean bandwidth of 4.6MBit/s and Gamma distributed request sizes for C-requests. We assumed that D-request sizes are Normal distributed with a mean of 200000 Bytes and a standard deviation of 50000. In contrast to our analytical model we did not restrict the waiting room for incoming D-requests (so that requests were never rejected) and we allowed requests to the same cylinder within the same round.

5.2 Results

We studied two scenarios: the first one with a high C-load and a low D-load and the second one with reversed roles. In both cases we varied the arrival rate λ_D of D-requests up to a maximum sustainable rate and compared the mean values $E[R]$ and the second moments $E[R^2]$ of the D-request response time that were measured in the simulations versus the analytical predictions. In the analytical model, the maximum queue length L_{max} was set to $3 * M$ to get a blocking probability below 0.1%. All symbolic computations (substitutions of equations, summations, etc.) were implemented in Maple.

In the first scenario, the maximum number of D-requests, M, served in one round was set to 8. The value for N, the number of C-requests that are served in each round, was also set to 8. The results of the simulations versus the analytical predictions are shown in **Table 3**.

λ_D $[s^{-1}]$	4	5	6	7
$E[R]$ $[s]$ (simulation)	0.8839	0.9185	0.9963	1.2384
$E[R]$ $[s]$ (analytic)	0.8857	0.9196	0.9950	1.2212
$E[R^2]$ $[s^2]$ (simulation)	0.9183	0.9913	1.1731	1.8968
$E[R^2]$ $[s^2]$ (analytic)	0.9225	0.9957	1.1715	1.8071

Table 3: Analytically Predicted vs. Measured Response Time ($M = 8$, $N = 8$)

λ $[s^{-1}]$	7	8	9	10	11	12	13	14
$E[R]$ $[s]$ (simulation)	0.8131	0.8285	0.8462	0.8656	0.8935	0.9374	1.0235	1.2850
$E[R]$ $[s]$ (analytic)	0.8135	0.8297	0.8477	0.8692	0.8980	0.9423	1.0267	1.2573
$E[R^2]$ $[s^2]$ (simulation)	0.7794	0.8083	0.8420	0.8805	0.9370	1.0319	1.2372	2.0631
$E[R^2]$ $[s^2]$ (analytic)	0.7828	0.8124	0.8460	0.8874	0.9446	1.0384	1.2381	1.8943

Table 4: Analytically Predicted vs. Measured Response Time ($M = 15$, $N = 5$)

In the high D-load scenario the maximum number of D-requests, M, was set to 15, and the number of C-requests served in each round, N, was set to 5. The results are shown in **Table 4**.

Both tables show that the predictions of the analytical model are highly accurate; the relative error is consistently in the order of one percent. The error becomes larger only when the arrival rate λ_D of D-requests approaches the saturation point of the disk (i.e., with the disk utilization becoming dangerously close to 100 percent). Overall the experimental results demonstrate that the developed analytical model is highly accurate for realistic cases.

In addition to the mean response time, we computed the probability that the response time of D-requests exceeds a given threshold using Chernoff's bound. **Table 5** shows the results for a response time threshold of 1.8 seconds with M set to 15 and N set to 5, i.e., the same values as used for the mean response time in Table 4. The analytical values are derived based on equation 32. They are compared with the results of our simulation. The goal is to bound the tail of the response time distribution such that less than 5 percent of the D-requests exhibit response times higher than 1.8 seconds. Using our analysis this goal can be achieved for a maximum arrival rate of 11 D-requests per second, whereas the simulation shows that an arrival rate of 13 D-requests per second can be sustained. So the analytic prediction conservatively stays below the true upper bound, but within reasonable limits.

6 Conclusion

The developed analytic performance-prediction model can form the basis of a capacity planning and server configuration tool. In particular, the model can be used for determining the minimum number of server disks, that are needed to meet the application's specified performance (i.e., D-request response time) and QoS (i.e., C-stream glitch rate) requirements.

Our current scheduling model is somewhat conservative as it considers a gated service for D-requests with a given limit on the number of D-requests served in a round. For low to moderate C-load, better response time of D-requests can be achieved by including them in the disk SCAN incrementally as they arrive (i.e., even within their arrival round) [15]. The analytic model developed in this paper yields a conservative bound with regard to the performance of such a dynamic and incremental scheduling policy. So the presented configuration method may overshoot a bit with regard to the required number of disks. More research is needed, however, to investigate if the more sophisticated scheduling policy is analytically tractable.

Although, in technical terms, our analytical model focuses on the required number of disks, the analysis implicitly considers also other factors in the overall server cost. The number of disks attached to a server determines how much expensive interconnect hardware like I/O boards are required to really exploit the disk I/O bandwidth, and ultimately, the server model that is needed. So minimizing the number of disks for given performance and QoS requirements of an application is not just for the sake of good engineering practice, but can lead to substantial cost savings in many installations of a multimedia information server.

References

[1] Fouad A. Tobagi, Joseph Pang, Randall Baird, and Mark Gang. Streaming RAID - A Disk Array Management System for Video Files. In *Proceedings of the ACM International Conference on Multimedia (ACM Multimedia'93), Anaheim, California*, August 1993.

[2] D. James Gemmell, Jiawei Han, Richard J. Beaton, and Stavros Christodoulakis. Delay-Sensitive Multimedia on Disks. *IEEE Multimedia*, pages 57–67, 1994.

$\lambda\ [s^{-1}]$	7	8	9	10	11	12	13	14
$P[R > 1.8s]$ (simulation)	$5.7E{-}5$	$2.13E{-}4$	$6.11E{-}4$	$1.70E{-}3$	$4.51E{-}3$	$1.35E{-}2$	$4.20E{-}2$	$1.67E{-}1$
$P[R > 1.8s]$ (analytic)	$4.42E{-}3$	$3.12E{-}3$	$7.76E{-}3$	$1.85E{-}2$	$4.39E{-}2$	$1.07E{-}1$	$2.75E{-}1$	$6.89E{-}1$

Table 5: Measured vs. Analytically Bounded Tail of Response Time Distribution

[3] Banu Özden, Rajeev Rastogi, and Avi Silberschatz. Disk Striping in Video Server Environments. In *Proceedings of the International Conference on Multimedia Computing and Systems (ICMCS'96), Hiroshima, Japan*, June 1996.

[4] Ed Chang and Avideh Zakhor. Admissions Control and Data Placement for VBR Video Servers. In *Proceedings of the 1st IEEE International Conference on Image Processing (ICIP'94), Austin, Texas*, pages 278–282, November 1994.

[5] Shahram Ghandeharizadeh, Seon Ho Kim, and Cyrus Shahabi. On Disk Scheduling and Data Placement for Video Servers. *ACM Multimedia Systems*, 1996.

[6] Harrick M. Vin, Pawan Goyal, Alok Goyal, and Anshuman Goyal. A Statistical Admission Control Algorithm for Multimedia Servers. In *Proceedings of the the ACM International Conference on Multimedia (ACM Multimedia '94), San Francisco, CA*, October 1994.

[7] Ed Chang and Avideh Zakhor. Variable Bit Rate MPEG Video Storage on Parallel Disk Arrays. In *Proceedings of the 1st International Workshop on Community Networking Integrated Multimedia Services to the Home, San Francisco, California*, pages 127–137, July 1994.

[8] Guido Nerjes, Peter Muth, and Gerhard Weikum. Stochastic Service Guarantees for Continuous Data on Multi-Zone Disks. In *Proceedings of the 16th Symposium on Principles of Database Systems (PODS'97), Tucson, Arizona*, pages 154–160, May 1997.

[9] Richard R. Muntz, Jose Renato Santos, and Steven Berson. A Parallel Disk Storage System for Realtime Multimedia Applications. *International Journal on Intelligent Systems, Special Issue on Multimedia Computing Systems*, 1998.

[10] Heiko Thimm, Wolfgang Klas, Crispin Cowan, Jonathan Walpole, and Calton Pu. Optimization of Adaptive Data-Flows for Competing Multimedia Presentational Database Sessions. In *Proceedings of the International Conference on Multimedia Computing and Systems (ICMCS'97), Ottawa, Canada*. IEEE, June 1997.

[11] Cliff Martin, P. S. Narayan, Banu Özden, Rajeev Rastogi, and Avi Silberschatz. The Fellini Multimedia Storage Server. In Soon M. Chung, editor, *Multimedia Information Storage and Management*. Kluwer, 1996.

[12] Leana Golubchik, John C.S. Lui, Edmundo de Silva e Souza, and H. Richard Gail. Evaluation of Tradeoffs in Resource Management Techniques for Multimedia Storage Servers. Technical Report TR-3904, University of Maryland, Department of Computer Science, 1998.

[13] Guido Nerjes, Peter Muth, and Gerhard Weikum. Stochastic Performance Guarantees for Mixed Workloads in a Multimedia Information System. In *Proceedings of the IEEE International Workshop on Research Issues in Data Engineering (RIDE'97), Birmingham, UK*, pages 131–140, April 1997.

[14] Guido Nerjes, Peter Muth, Michael Paterakis, Yannis Romboyannakis, Peter Triantafillou, and Gerhard Weikum. On Mixed-Workload Multimedia Storage Servers with Guaranteed Performance and Service Quality. In *Proceedings of the 3rd International Workshop on Multimedia Information Systems (MIS'97), Como, Italy*, pages 63–72, October 1997.

[15] Guido Nerjes, Peter Muth, Michael Paterakis, Yannis Romboyannakis, Peter Triantafillou, and Gerhard Weikum. Scheduling Strategies for Mixed Workloads in Multimedia Information Servers. In *Proceedings of the IEEE International Workshop on Research Issues in Data Engineering (RIDE'98), Orlando, Florida*, February 1998.

[16] Yannis Romboyannakis, Guido Nerjes, Michael Paterakis, Peter Triantafillou, and Gerhard Weikum. Disk Scheduling for Mixed-Media Workloads in a Multimedia Server. In *Proceedings of the ACM International Conference on Multimedia (ACM Multimedia'98), Bristol, UK*, September 1998.

[17] Hideaki Takagi. *Queueing Analysis : A Foundation of Performance Analysis, Volume 1 : Vacation and Priority Systems*. North-Holland, Amsterdam, 1991.

[18] Edward G. Coffman Jr. and Micha Hofri. Queueing Models of Secondary Storage Devices. In Hideaki Takagi, editor, *Stochastic Analysis of Computer and Communication Systems*. North-Holland, 1990.

[19] Arnold O. Allen. *Probability, Statistics and Queueing Theory with Computer Science Applications*. Academic Press, 2nd edition, 1990.

[20] Randolph Nelson. *Probability, Stochastic Processes, and Queueing Theory : The Mathematics of Computer Performance Modeling*. Springer, 1995.

[21] Leonard Kleinrock. *Queueing Systems, Volume 1: Theory*. Wiley, 1975.

[22] Philip S. Yu, Mon-Song Chen, and Dilip D. Kandlur. Grouped Sweeping Scheduling for DASD-based Multimedia Storage Management. *ACM Multimedia Systems*, 1(3):99–109, 1993.

[23] Steven Berson, Shahram Ghandeharizadeh, Richard R. Muntz, and Xiangyu Ju. Staggered Striping in Multimedia Information Systems. In *Proceedings of the International Conference on Management of Data (SIGMOD'94), Minneapolis, Minnesota*, pages 79–90, May 1994.

[24] D. James Gemmel, Harrick M. Vin, Dilip D. Kandlur, P. Venkat Rangan, and Lawrence A. Rowe. Multimedia Storage Servers: A Tutorial. *IEEE Computer*, pages 40–49, May 1995.

[25] Oliver Rose. Statistical Properties of MPEG Video Traffic and Their Impact on Traffic Modeling in ATM Systems. Technical Report Technical Report 101, Institute of Computer Science, University of Würzburg, Germany, 1995.

[26] Marwan Krunz and Herman Hughes. A Traffic Model for MPEG-Coded VBR Streams. In *Proceedings of the International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'95), Ottawa, Canada*, May 1995.

[27] Chris Ruemmler and John Wilkes. An Introduction to Disk Modeling. *IEEE Computer*, 27(3):17–28, March 1994.

[28] Bruce L. Worthington, Gregory R. Ganger, Yale N. Patt, and John Wilkes. On-Line Extraction of SCSI Disk Drive Parameters. In *Proceedings of the International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'95), Ottawa, Ontario, Canada*, May 1995.

[29] Yen-Jen Oyang. A Tight Upper Bound of the Lumped Disk Seek Time for the Scan Disk Scheduling Policy. *Information Processing Letters*, 54:355–358, 1995.

[30] Torben Hagerup and Christiane Rüb. A Guided Tour of Chernoff Bounds. *Information Processing Letters*, 33:305–308, 1989.

[31] Tony T. Lee. M/G/1/N Queue with Vacation Time and Limited Service Discipline. *Performance Evaluation*, 9:181–190, 1989.

Appendix: List of Used Symbols

A	random variable for the time a D-request spends until its arrival round ends
C	duration of a single stream
D	random variable for the time a D-request spends in its service round
I	random variable for the time a D-request spends until its service round begins
L_{t_i}	queue length at beginning of round i
L	queue length at arbitrary point of time
L_{max}	maximum length of D-request waiting queue
M_D	maximum number of D-requests served in a single round
M_C	constant number of C-requests served in each round per disk
N	number of C-requests per round
N_{max}	maximum number of streams such that QoS can be guaranteed
$O(t)$	random variable for the number of D-request arrivals in time period of length t
R	random variable for the response time of a D-request
S	random variable for size of fragments
T	round length = playback time of a single fragment
T_N	total service Time for the retrieval of N fragments
$T_{rot,i}$	rotational delay of i-th request in round
$T_{seek,i}$	seek time of i-th request in round
$T_{trans,i}$	transfer time of i-th request in round
T_{seek}	upper-bound of all seek times in a round
A_l^*	Laplace transform of joint probability $P[L = l \wedge A = t]$
B_D^*	Laplace transform of rotational delay plus transfer time for a D-request
B_C^*	Laplace transform of rotational delay plus transfer time for a C-request
$B_{scan1,n}^*$	Laplace transform of SCAN seek time for first seek with n requests in SCAN
$B_{scanX,n}^*$	Laplace transform of SCAN seek time with n requests in SCAN
B_{rot}^*	Laplace transform of rotational delay
$B_{read_D}^*$	Laplace transform of transfer time for a D-request
$B_{read_C}^*$	Laplace transform of transfer time for a C-request
D_l^*	Laplace transform of conditional probability $P[D = x \mid L = l \wedge A = t]$
I_l^*	Laplace transform of conditional probability $P[I = x \mid L = l \wedge A = t]$
R^*	Laplace transform of R
T_N^*	Laplace transform of T_N
a_k	probability for k arrivals during a round of length T
b_{error}	chernoff's bound on the probability p_{error}
b_{glitch}	bound of probability p_{glitch}
b_{late}	chernoff's bound on the tail probability p_{late}
c_{cyl}	number of disk cylinders
c_{min}	minimum track capacity for multi-zone disk
c_{max}	maximum track capacity for multi-zone disk
c_{rot}	maximum rotational delay
c_{trans}	constant transfer rate of single-zone disk
c_z	number of zones for multi-zone disk
f_{rate}	density of transfer rate for multi-zone disk
$f_{scan1,n}$	density of seek distance distribution for first seek in SCAN with n requests
$f_{scanX,n}$	density of seek distance distribution for subsequent seeks in SCAN with n requests
f_{size_D}	density of D-request size distribution
f_{size_C}	density of C-request size distribution
P_B	blocking probability
p_{error}	probability that a stream suffers a glitch during a single round
p_{glitch}	probability that a stream suffers a glitch during a single round
p_{late}	tail probability of the random variable T_N
p_n	steady state probability $P[L_{t_i} = n]$
q_i	probability for serving i D-requests in a round
t_i	time point at beginning of round i
$seek(d)$	seek time function, calculates seek time for a given seek distance d
$seek^{-1}(t)$	inverse seek time function, calculates seek distance for a given seek time t
λ	arrival rate of D-requests per disk