Stochastic Service Guarantees for Continuous Data on Multi-Zone Disks*

Guido Nerjes
Swiss Federal Institute of Technology (ETH)
Institute of Information Systems
CH-8092 Zurich, Switzerland
nerjes@inf.ethz.ch
http://www-dbs.inf.ethz.ch

Peter Muth
University of the Saarland
Department of Computer Science
D-66041 Saarbrücken, Germany
muth@cs.uni-sb.de
http://www-dbs.cs.uni-sb.de

Gerhard Weikum
University of the Saarland
Department of Computer Science
D-66041 Saarbrücken, Germany
weikum@cs.uni-sb.de
http://www-dbs.cs.uni-sb.de

Abstract

Continuous data types like video and audio require the real-time delivery of data fragments from a server's disks to the client at which the data is displayed. This paper develops a stochastic model for analyzing the rate at which data fragments arrive too late at the client and thus cause display "glitches". The model is based on deriving the Laplace-Stieltjes transform of the service time distribution for batched disk service under a multi-user load of concurrently served continuous-data streams, and applying Chernoff bounds to the tail of the service time distribution and the resulting distribution of the glitch rate per stream. The results from the model provide the basis for configuring a server and exerting an admission control such that the admitted streams suffer no more than a specified (small) rate of glitches with a specified (very high) probability. The model considers variable display bandwidth both across different streams and within a single stream, and also the variable transfer rate of modern multi-zone disks. The accuracy of the model is validated by detailed simulations.

1 Introduction

Multimedia applications such as news-on-demand or teleteaching pose challenging performance demands on the underlying storage servers [GVK95, Chu96, SJ96]. A well known requirement is that continuous data like video and audio calls for performance guarantees in terms of data delivery time to ensure "hiccup-free" display at the client site. Such guarantees can be deterministic (e.g., [BGM94, GK95, ÖRS96, VGG95]), leaning towards the worst case, or stochastic in that the probability for degraded service quality is limited [CZ94, VGG94, CL96]. For example, the deterministic approach would ensure that data "glitches" (i.e., data portions that do not arrive in time) do never occur, whereas a stochastic approach would bound the glitch probability within one data stream by say one percent.

Deterministic guarantees are mandatory for special, extremely critical applications with a small number of concurrent clients such as tele-surgery. However, they are feasible only if the resource demands of the concurrently served data streams are modeled conservatively, usually based on simplifying assumptions such as constant (maximum) and uniform display bandwidth of data objects, and constant seek time and rotational latency for the server's disks. By properly interpreting these performance factors as random variables with a certain distribution function, a stochastic approach typically leads to a much better resource utilization at the expense of a certain, very small probability that resources are overcommitted.

In Proceedings of the 16th Symposium on Principles of Database Systems (PODS'97), Tucson, Arizona, May 1997

In this paper, we pursue stochastic guarantees for continuous-data requests. Our objective is to bound the probability that data portions are behind their delivery deadline according to the real-time display requirements. Being able to predict this glitch probability can be exploited for configuring the server (choosing the number of disks, etc.) and, most importantly, the admission control of the server. Our approach is based on coarse-grained striping of the data across the server's disks and a disk scheduling scheme that operates in *rounds*, which is the prevalent scheduling approach for video servers [TPBG93, BGM94, CKY93, GHB94, VGG95, ÖRS96, CZ96]. When a client request arrives to open a new continuous-data stream while the server is already serving N ongoing streams, the new stream is admitted only if the predicted glitch probability for the new multiprogramming level $N+1$ does not exceed a specified tolerance threshold. Otherwise the request is turned away or postponed until one or more active streams terminate.

Technically, the stochastic service quality approach is based on an analytic model for the distribution of the total service time per round. We first derive the Laplace-Stieltjes transform of this distribution, and then determine the distribution of the number of glitches per round. Chernoff bounds [Kle75, Nel95] are applied to limit the tail of these distributions. Further probabilistic considerations finally yield the desired bound for the number of glitches over all rounds of one stream. A technical challenge in the first step is to accurately model the behavior of modern multi-zone disks [RW94, TCG96a]. Such disks group adjacent tracks into zones and use different numbers of sectors per track on inner and outer zones in order to fully utilize the raw disk capacity and bandwidth.

To the best of our knowledge, this is the first paper that develops a reasonably accurate analytic model for the service quality of a multi-user continuous-data server with *multi-zone disks* and *variable-bandwidth data objects*. Prior work is mostly based on conventional, single-zone disks and assumes constant seek time and rotational latency as well as constant-bandwidth objects in their models (e.g., [BGM94, VGG95, ÖRS96]), or resort to computationally expensive simulations of the exact disk behavior [CZ96] that seem to be infeasible for run-time admission control decisions. The only work modeling disk seek times and disk transfer times as random variables are [CZ94, VGG94, CL96], and our recent work [NMW97]. However, with the exception of [NMW97], their stochastic guarantees assume independent disk seeks rather than the more efficient SCAN policy for the disk arm positioning, and they rely on the applicability of the central limit theorem which is often questionable in practice. Also, none of these models considers multi-zone disks.

Our own prior work [NMW97] has focused on studying the performance impact of continuous-data streams on the service of conventional, "discrete" data requests in a mixed-workload multimedia server. To this end, [NMW97] has developed a stochastic model for

* This work has been supported by the ESPRIT Long Term Research Project No 9141, HERMES (Foundations of High Performance Multimedia Information Management).

the batched disk service of concurrent continuous-data streams. The current paper extends this model significantly in that it uses a more realistic model for the size distribution of compressed-video data fragments and derives much tighter bounds for the probability distribution of the glitch rate on a per stream basis. In addition and most notably, the current paper analyzes the influence of modern multi-zone disks, which has been disregarded in [NMW97]. Studies of the performance impact of multi-zone disks on continuous-data servers, on the other hand, have focused on data placement issues, namely, the layout of constant-bandwidth data objects across zones [Bir95, GKS96] and the generalization of organ-pipe-like arrangements [TKKD96, TCG96b].

The rest of the paper is organized as follows. Section 2 introduces our system architecture. Section 3 develops an analytic model for predicting the glitch probability. In Section 4, the approach is validated by comparing the analytic results to simulation studies. Section 5 discusses some practical system issues, and Section 6 gives an outlook on future work. A prototype system based on this new approach is being built by extending the FIVE experimental file system [SWZ94, SWZ96].

2 System Architecture

We assume that clients submit requests for continuous data to the server. Continuous-data objects like videos, audios, or animations are composed of sequences of *fragments* and constitute data *streams* that are consumed by the client in a time-constrained way according to the display bandwidth of the object. We assume that each client provides a certain amount of memory as a buffer for incoming fragments. The buffer size may vary among clients according to the local resources available. The buffer size must not be below a certain minimum allowing the server to deliver a fragment before the previous one is consumed by the client. We assume a fast and reliable network with a performance capacity well above our bandwidth requirements, and thus disregard network issues in this paper.

2.1 Data Layout

We consider a single server with D disks. Since video and audio compression techniques reduce the bandwidth of videos and audios substantially, we assume that the display bandwidth $r_{display}$ of a continuous object is always smaller than the bandwidth r_{disk} of a single disk.

Fragments are assigned to disks in a round robin fashion, similarly to the coarse-grained striping approach of [ÖRS96] and the simple/staggered striping approach of [BGM94] specialized to the case with cluster size 1 and stride 1. The salient properties that we share with these approaches are twofold:

- The load is balanced across the disks, assuming that objects are sufficiently large to be spread across all disks and that most users consume complete objects (as opposed to fast-forwarding a video or viewing only a short prefix).
- The server can sustain more concurrent (but time-wise unrelated) streams on the same object than it would be possible by multiplexing the service of a single disk in case the entire object resided on this disk. With D disks, disk bandwidth r_{disk}, and object display bandwidth $r_{display}$, the server can, in principle, support up to $D * r_{disk} / r_{display}$ streams on the same or different objects, under the optimistic and usually unrealistic assumption that multiplexing does not lead to a reduction of the effective disk bandwidth. (This consideration is for illustration only; the model of Section 3 does consider the reduction of the effective disk bandwidth.)

We consider data objects with variable display bandwidth (also known as VBR = variable bit rate objects), as compression techniques such as MPEG-2 result in a variable bandwidth over time. In our scheme, all data fragments stored by the server have the same display time [CZ96, CBR95], i.e. the time it takes a client to consume a fragment (e.g., a few seconds). As a consequence, fragments vary in size. By "normalizing" all fragments to the same time length, we induce a periodic access pattern with a uniform period across all continuous objects regardless of the display bandwidth differences between objects and the variation within an object (the latter being due to compression). This type of fragmentation requires parsing a continuous object before it is laid out on the server's disks, but this is straightforward and inexpensive given that, in most applications, continuous objects are never modified after their initial insertion.

2.2 Multi-zone Disks

Conventional disks have a fixed number of sectors per track. Thus, data on inner tracks is stored at a higher areal density than data on outer tracks. By storing all data at the same, highest possible areal density, the capacity of outer tracks can be increased. Multi-zone disks exploit this idea by grouping adjacent tracks into *zones*, and allowing more sectors per track in the outer zones than in the inner zones [RW94]. As the angular velocity is kept constant, this approach also yields a higher transfer rate for the outer zones. Typical high performance disks have a capacity and transfer rate ratio between outer and inner tracks of a factor of two. This is clearly an important performance factor, especially with the relatively large request sizes for continuous data (which are in the order of 50 to a few hundred KBytes).

In this paper, we assume that information about zones is not used for the placement of data on disk, i.e., the data is uniformly distributed over all sectors of the disk. More advanced placement schemes with information about access frequencies should employ a generalized organ-pipe permutation [Won83], storing the hottest data at an optimal point somewhere between the middle and the outermost track [TKKD96, TCG96b], to find the best compromise between short seeks and high bandwidth. Taking such placement optimizations into consideration is left for future work. Note that we assume the bandwidth requirements of all requests to be less than the bandwidth of the innermost zone, such that any data can be stored on any zone.

2.3 Admission Control and Disk Scheduling

The scheduling consists of a global admission control and a separate disk scheduler for each disk. Initialization requests for opening a new continuous-data stream have to pass the admission control first. Only a limited number of concurrent streams can be sustained. Therefore, the admission control rejects new initialization requests when the server load becomes too high. Workload statistics, e.g., on the distribution of fragment sizes, are fed into the admission control.

Now consider the actual disk scheduling. The periodic pattern of the requests for the admitted streams suggests a cyclic scheduling scheme that proceeds in rounds, with a round length t equal to the display time of a fragment, which is uniform across all fragments. The round length is a configuration parameter of our architecture; changing it would require all data to be re-fragmented. During a round, all requests of the admitted streams have to be served (the order is arbitrary, the corresponding fragments will be displayed by the clients in the subsequent round). Given that a fragment always resides on a single disk, there are no dependencies among the requests of one round, so that we can schedule the requests of each disk separately, as long as we complete all requests by the end of the round. In order to minimize disk seeks, we use the SCAN algorithm for the disk arm movement [SG94] (also known as the 'elevator' algorithm). With this algorithm, all requests of one round are sorted according to their seek position on the disk and are served with one sweep of the disk arm.

Our approach to find the maximum number of concurrent streams that can be allowed by the admission control is based on a stochastic model. Given the length t of a scheduling round, we derive the maximum number, N_{max}, of concurrent data streams that can be

served during a single round such that the probability for one stream suffering more than a certain tolerable number of glitches in a certain number of successive rounds stays below a specified threshold, say one percent. The computed value of N_{max} is then used to drive the admission control in that only up to N_{max} streams can be admitted. An admitted stream may receive a small startup delay of up to one round's duration; however, given that a round is relatively short (a few seconds), we are not concerned with this aspect.

3 Analytic Model

In this section we develop a stochastic model that allows us to limit the glitch probability within one stream of M rounds. We proceed in three steps. In Section 3.1 we first derive a Chernoff bound [Kle75, Nel95] for the tail of the probability distribution of the total service time per round, assuming N concurrent streams on a conventional, single-zone disk. In Section 3.2 we then extend this result to multi-zone disks by additionally considering the probability distribution of the transfer time for one fragment. Finally, Section 3.3 uses the derived Chernoff bounds to develop a bound for the glitch probability of an individual stream. Throughout the section we consider only one disk and its corresponding load, which is feasible as there are no scheduling dependencies among different disks (see Section 2). Thus, all workload parameters, particularly, the multiprogramming level N, are on a per disk basis, assuming that the load is uniformly distributed across disks.

3.1 Total Service Time Per Round With Conventional Disks

The key problem to be solved here is to estimate the total service time for the N requests of one round, using a SCAN policy for the disk arm movement. Prior work on this problem used constant worst case values for the seek and rotational delays between successive data transfers, or assumed that the total (i.e., accumulated) seek time of one sweep over the disk equals the maximum seek time of the disk. This yields a deterministic but unrealistic estimate since it ignores the stochastic nature of rotational delays and the non-linearity of the disk arm movement [RW94]. The only work addressing this problem by a profound stochastic model are [CL96] and [CZ94]. [CL96] assumes independent seeks for the N requests rather than the much better SCAN policy, and arrives at a relatively coarse bound based on the Tschebyscheff inequality. [CZ94] is also based on independent seeks and assumes that N is sufficiently large to apply the central limit theorem (i.e., consider only the limit $N \to \infty$) and thus assume that the total service time is normally distributed, which is not always justified for realistic values of N (e.g., 10 to 50 streams per disk). In the following we derive a much more accurate stochastic model and a much tighter bound using a recent result on the total seek time of the SCAN policy [Oya95] and the method of Chernoff bounds [Kle75, Nel95].

Let T_N denote the total service time for a round with N requests. Then we have

$$T_N = T_{seek} + \sum_{i=1}^{N} T_{rot,i} + \sum_{i=1}^{N} T_{trans,i} \qquad (3.1.1)$$

where T_{seek} is the accumulated seek time for one sweep of the SCAN policy, $T_{rot,i}$ is the rotational delay and $T_{trans,i}$ is the transfer time of the ith request.

According to [Oya95] T_{seek} is maximized, under a realistic function for the seek time, for equidistant seek positions of the N requests. The seek time function itself is assumed to be proportional to the square root of the seek distance for small distances below a disk-specific constant, and a linear function of the seek distance for longer distances, which is in accordance with the studies of [RW94]. Thus, for given disk parameters, the maximum total seek time of a sweep can be easily computed by assuming the N seek positions to be at cylinders $i*CYL / (N+1)$ for $i=1, ..., N$ where CYL is the total number of the disk's cylinders, and applying the seek time function. This computation yields an upper bound for T_{seek} which, other than depending on N, can now be viewed as a constant, denoted $SEEK$ in the following.

The N random variables $T_{rot,i}$ are independently and identically distributed with a uniform distribution between 0 and the time for one disk revolution, ROT. Similarly, the random variables $T_{trans,i}$ are independently identically distributed. This distribution depends on the distribution of data fragments and the disk's transfer rate (which in turn is a function of the revolution speed and the head switch time). Based on statistical studies of the size distribution of compressed-video data fragments [Ros95, KH95], we assume that $T_{trans,i}$ has a Gamma distribution [All90] with a mean value $E[T_{trans,i}]$ and variance $Var[T_{trans,i}]$. This distribution captures the variability of request sizes, which in turn is due to the bandwidth variation within and across objects, and is substantially more realistic than assuming constant request sizes as it is done in most of the prior work. Note that the following derivation can be carried out also with other distributions of the data fragment size (i.e., other heavy-tailed distributions such as Pareto or Lognormal) as long as we can derive (or approximate) the corresponding Laplace-Stieltjes transform.

So T_{seek} is equal to the constant $SEEK$, and the probability density functions of $T_{rot,i}$ and $T_{trans,i}$ are given by (Γ denotes the Gamma function):

$$f_{rot}(x) = \frac{1}{ROT} \quad \text{and} \quad f_{trans}(x) = \frac{\alpha(\alpha x)^{\beta-1}\, e^{-\alpha x}}{\Gamma(\beta)}, \qquad (3.1.2)$$

$$\text{with } \alpha = \frac{E[T_{trans,i}]}{Var[T_{trans,i}]} \text{ and } \beta = \frac{(E[T_{trans,i}])^2}{Var[T_{trans,i}]},$$

and their Laplace-Stieltjes transforms [Fel71, Kle75, Nel95] are given by

$$T^*_{seek}(s) = e^{-s\,SEEK}, \quad T^*_{rot,i}(s) = \frac{1 - e^{-s\,ROT}}{s\;ROT}, \quad \text{and}$$

$$T^*_{trans,i}(s) = \left(\frac{\alpha}{\alpha + s}\right)^{\beta}. \qquad (3.1.3)$$

The Laplace-Stieltjes transform of T_N, which involves the convolution of the N-fold convolution of $T_{rot,i}$ and $T_{trans,i}$, is given by

$$T^*_N(s) = e^{-s\,SEEK} \left(\frac{1 - e^{-s\,ROT}}{s\;ROT}\right)^N \left(\frac{\alpha}{\alpha + s}\right)^{\beta * N} \qquad (3.1.4)$$

and the corresponding moment generating function $M(s)$ equals $T^*_N(-s)$. Now we are ready to apply Chernoff's theorem to bound the tail of the random variable T_N. Namely, the following inequation holds [Kle75, Nel95]:

$$P[T_N \geq t] \leq inf_{\theta \geq 0} \left\{ e^{-\theta t}\, M(\theta) \right\} = inf_{\theta \geq 0} \{ h(\theta) \} \text{ with}$$

$$h(\theta) = e^{-\theta t}\, e^{\theta\, SEEK} \left(\frac{e^{\theta\, ROT} - 1}{\theta\; ROT}\right)^N \left(\frac{\alpha}{\alpha - \theta}\right)^{\beta * N}. \qquad (3.1.5)$$

For the given form of h, differentiating h and solving $h' = 0$ for θ yields the optimum value of θ to obtain the sharpest bound in the Chernoff inequation. While we did not manage to obtain a closed form expression for this result, solving $h' = 0$ numerically is straightforward and very efficient.

So finally, when we consider a round of fixed duration t, the probability $p_{late}(N,t)$ for not being able to serve the requests of all N streams within one round of duration t and its bound $b_{late}(N,t)$ are obtained by

$$p_{late}(N,t) = P[T_N \geq t] \leq h(\textit{solution of } h' = 0) = b_{late}(N,t) \qquad (3.1.6)$$

For example, consider a round length $t = 1$ second and data fragment sizes with a mean of 200 KBytes and a standard deviation of 100 KBytes which results in $E[T_{trans,i}] = 0.02174$s and $Var[T_{trans,i}] = 0.00011815\text{s}^2$ for a disk with a track capacity of 75 KBytes and rotation time $ROT = 0.00834$ seconds. For this disk and $N = 27$, we obtain $SEEK = 0.10932$ seconds, and the derived upper

bound for p_{late} is approximately 0.0103. In other words, we can guarantee with a probability of at least *1 - p_{late}* ≈ 0.9897 that all N=27 requests of one round can be served within the period of length t = 1 second. If our goal is to guarantee the timely service of N requests with a probability of at least 0.99, then we have to (slightly) lower the value of N. For N=26 we obtain $p_{late} \approx 0.00225$, and this would achieve the goal. In general, for a given value of t and a threshold δ for p_{late}, we can derive the maximum number of admissible concurrent streams as

$$N_{\max}^{plate} = \max\left\{N \mid p_{late}(N,t) \leq \delta\right\}. \quad (3.1.7)$$

3.2 Considering Multi-Zone Disks

The additional performance impact incurred by multi-zone disks is due to variable track capacity and the variable transfer rate. If we assume that the requested fragments are allocated uniformly across the sectors of a disk, the variable track capacity induces a skewed distribution for the tracks that are selected by the requests, with a higher probability of outer tracks. As for seek times, this increases the probability for shorter seeks. Note, however, that the worst-case bound from [Oya95] that we have used in Section 3.1 is valid for multi-zone disks, too, and we can again adopt this bound for the following analysis. This simplification is justified as the skewed seek time distribution has certainly much less of an impact than the variable transfer rate which may vary by a factor of two between the innermost and the outermost tracks. Thus, we concentrate on modeling the impact of multi-zone disks on the distribution of the transfer time.

Consider a multi-zone disk with Z zones, numbered 1 through Z from the innermost zone to the outermost zone, with zone i having a per track storage capacity of C_i and a transfer rate $R_i = C_i / ROT$. Denote the capacity of the innermost zone by C_{min} and that of the outermost zone by C_{max}. Under the assumption that all zones have the same number of tracks, the probability for a request to hit zone i is $\frac{C_i}{C}$ with $C = \sum_{i=1}^{Z} C_i$. Thus, the probability for a request to be served with transfer rate $R \leq R_i$ is:

$$P[\text{transfer rate } R \leq R_i] = \sum_{j=1}^{i} C_j / C. \quad (3.2.1)$$

By assuming that the track capacity of the zones and thus their transfer rate increase linearly, C_i and R_i are given by

$$C_i = C_{\min} + \frac{(C_{\max} - C_{\min}) * (i-1)}{Z-1} \quad \text{for } i=1,...,Z \quad (3.2.2)$$

$$R_i = \left(C_{\min} + \frac{(C_{\max} - C_{\min}) * (i-1)}{Z-1}\right) * \frac{1}{ROT} \quad \text{for } i=1,...,Z \quad (3.2.3)$$

Substituting C_i into (3.2.1) leads to

$$P[\text{transfer rate } R \leq R_i] = \frac{i\, C_{\min} + \frac{C_{\max} - C_{\min}}{Z-1}\frac{(i-1)i}{2}}{C}, \quad (3.2.4)$$

and setting $r := R_i$, solving (3.2.3) for i, and substituting this value of i into the right hand side of (3.2.4) finally yields the distribution function $F_{rate}(r)$ for the random variable R:

$$F_{rate}(r) = P[R \leq r]$$
$$= \frac{(C_{\min}/ROT + r) * (r - Zr + ZC_{\min}/ROT - C_{\max}/ROT)}{(C_{\min} + C_{\max})Z(C_{\min} - C_{\max})/ROT^2} \quad (3.2.5)$$

In the following we will treat this distribution as if the transfer rate R were a continuous random variable, a standard approximation technique for analyzing discrete distributions. Thus, differentiating (3.2.5) by r yields the density function $f_{rate}(r)$ of the transfer rate:

$$f_{rate}(r) = \frac{2rZ - 2r + C_{\max}/ROT - C_{\min}/ROT}{(C_{\min} + C_{\max})Z(C_{\min} - C_{\max})/ROT^2} \quad (3.2.6)$$

Now consider the random variable T_{trans} that denotes the transfer time of a request (i.e., one fragment). This is the quotient of the request size and the transfer rate. Given a density function $f_{size}(x)$ for the distribution of the request size S and the above density function $f_{rate}(r)$, the density function of the transfer time, $f_{trans}(t)$, is the following convolution-like integral [Fel71]:

$$f_{trans}(t) = \int_{r=C_{\min}/ROT}^{C_{\max}/ROT} f_{rate}(r) * r * f_{size}(t * r)\, dr, \quad (3.2.7)$$

which specializes for Gamma distributed requests sizes with mean $E[S]$ and variance $Var[S]$ (as we assumed in Section 3.1) into a function of the following form:

$$f_{trans}(t) = \sum_{i=0}^{5} t^{(i-3)} * \left(c_i * e^{f_0 t} - d_i * e^{f_1 t}\right) \quad (3.2.8)$$

with positive constants c_0 through c_5, d_0 through d_5, f_0, and f_1 that depend on C_{min}, C_{max}, Z, ROT, $E[S]$ and $Var[S]$.

The next step would be to compute the Laplace-Stieltjes transform of f_{trans} which could then be plugged into the derivation of Section 3.1. Unfortunately, the integral $T^*_{trans}(s) = \int_{t=0}^{\infty} e^{-st} f_{trans}(t)\, dt$ cannot be solved in closed form for the obtained form of $f_{trans}(t)$. Therefore, we resort to approximating $f_{trans}(t)$ by a Gamma distribution with a density of the form $\frac{\alpha(\alpha x)^{\beta-1}e^{-\alpha x}}{\Gamma(\beta)}$, where the parameters α and β are determined such that the first two moments are identical to those of the actual distribution $f_{trans}(t)$. Numerical studies with typical values of the underlying constants C_{min}, C_{max}, Z, ROT, and $E[S]$ and $Var[S]$ show that the relative error of the approximation is less than 2 percent in the most relevant range of the transfer time (which is for t between 5 and 100 milliseconds, i.e., between half a disk revolution and about ten revolutions, given typical disk and data characteristics). Denote the approximated density function of the transfer time by $f_{apptrans}(t)$ and its Laplace-Stieltjes transform by $T^*_{apptrans}(s)$ where

$$T^*_{apptrans}(s) = \left(\frac{\alpha}{\alpha + s}\right)^{\beta} \quad (3.2.10)$$

Now we can proceed analogously to Section 3.1 and construct the Laplace-Stieltjes transform of the total service time for one round with N concurrent streams as the convolution of T^*_{seek}, the N-fold convolution of T^*_{rot}, and the N-fold convolution of $T^*_{apptrans}$:

$$T^*_N(s) \approx T^*_{seek}(s)\ (T^*_{rot}(s))^N\ (T^*_{apptrans}(s))^N, \quad (3.2.11)$$

Finally, the derivation of Chernoff bounds is analogous to Section 3.1 as well, and we obtain:

$$p_{late}(N,t) = P[\text{total service time for N streams} \geq t]$$
$$\leq \inf_{\theta \geq 0}\left\{e^{-\theta t}\ T^*_N(-\theta)\right\}$$
$$= b_{late}(N,t). \quad (3.2.12)$$

Again, the sharpest Chernoff bound is obtained by numerically determining the minimum (or, more generally, infimum) of $e^{-\theta t}\ T^*_N(-\theta)$ in (3.2.12). For example, for the disk and data characteristics given in Table 1 of Section 4, a round length of t = 1 second, and a multiprogramming level of N = 26, the probability p_{late} for not being able to serve all N requests within the round is at most 0.00324. Setting N = 27, on the other hand, would lead to a p_{late} value of 0.0133. Thus, if the goal is to limit the probability of one round being late by 1 percent, then N = 26 is the maximum admissible number of concurrent streams.

3.3 Glitch Probability

We first consider the distribution of the number of glitches in one round, and then derive the number of glitches affecting one stream that extends over M rounds. When we focus on one stream, we assume that the streams that are affected by glitches (i.e., miss one

fragment or have one fragment delayed) are selected independently among the rounds. Given that time-wise successive fragments of the same stream reside on different disks (see Section 2), this independence can be easily ensured by allocating fragments on their disk in a random manner. (One has to ensure that all fragments of one object reside in uncorrelated positions of the sweeps of the different disks). Under this natural condition, we can interpret k glitches in one round as a random drawing of k out of N streams. Thus, we obtain the probability that a particular stream is affected by a glitch as:

$$P[\textit{stream i has a glitch in one round}] = \sum_{k=1}^{N} \left(P[\textit{number of glitches per round} = k] * \frac{k}{N} \right) \quad (3.3.1)$$

Let T_k denote the total service time for k requests in a single round. Then equation (3.3.1) can be rewritten as:

$$\begin{aligned} &P[\textit{stream i has a glitch in one round}] \\ &= \frac{1}{N} \sum_{k=0}^{N-1} \left((N-k)\, P[\textit{ number of served streams} = k\ ] \right) \\ &= \frac{1}{N} \sum_{k=0}^{N-1} \left((N-k) \left(P[T_k \le t] - P[T_{k+1} \le t] \right) \right) \\ &= 1 - \frac{1}{N} \sum_{k=1}^{N} P[T_k \le t] \\ &= \frac{1}{N} \sum_{k=1}^{N} p_{late}(k,t) \end{aligned} \quad (3.3.2)$$

The probability $p_{late}(k,t)$ for not being able to serve all k requests within one round of duration t can be limited by the Chernoff bound derived in Section 3.2, equation (3.2.12). Thus there is a bound for the probability that a single stream suffers from a glitch in one round:

$$\begin{aligned} p_{glitch}(N,t) &= P[\textit{stream i has a glitch in one round}] \\ &\le \frac{1}{N} \sum_{k=1}^{N} b_{late}(k,t) \\ &= b_{glitch}(N,t) \end{aligned} \quad (3.3.3)$$

Then, the probability for one stream suffering g glitches in M rounds is

$$P[\textit{stream i has g glitches in M rounds}] = \binom{M}{g} * \left(p_{glitch}(N,t) \right)^{g} * \left(1 - p_{glitch}(N,t) \right)^{(M-g)} \quad (3.3.4)$$

This is a binomial distribution with parameters M and p_{glitch}; thus, evaluating formula (3.3.4), albeit feasible, would be computationally expensive. However, for the important case of a binomial distribution, the following efficiently evaluable Chernoff bound, derived in [HR89], can be applied for $g / M > p_{glitch}(N,t)$:

$$\begin{aligned} &p_{error}(N,t,M,g) \\ &= P[\textit{number of glitches of stream i in M rounds} \ge g] \\ &\le \left(\frac{M\, p_{glitch}(N,t)}{g} \right)^{g} \left(\frac{M - M\, p_{glitch}(N,t)}{M-g} \right)^{M-g} \\ &\le \left(\frac{M\, b_{glitch}(N,t)}{g} \right)^{g} \left(\frac{M - M\, b_{glitch}(N,t)}{M-g} \right)^{M-g} \end{aligned} \quad (3.3.5)$$

For example, for the disk and data characteristics given in Table 1 of Section 4, a multiprogramming level of $N = 28$, a round length of $t = 1$ second, and streams with $M = 1200$ rounds, the probability that an individual stream suffers more than 12 glitches (i.e., 1 percent of M) is at most $0.14*10^{-3}$.

Analogously to (3.1.7), for a given value of t, M, g, and a threshold ϵ for p_{error}, we can derive the maximum number of concurrent streams as

$$N_{\max}^{perror} = \max \{ N \mid p_{error}(N,t,M,g) \le \epsilon \} \quad (3.3.6)$$

4 Model Validation

To validate the developed analytic model, we compared the predictions of the model with results obtained from detailed simulations, using the characteristics given in Table 1. The disk parameters correspond to a Quantum Viking 2.1 drive.

number of cylinders	*CYL*	*6720*
number of zones	*Z*	*15*
revolution time	*ROT*	*8.34 ms*
track capacity of innermost zone	C_{min}	*58368 Bytes*
track capacity of outermost zone	C_{max}	*95744 Bytes*
seek time $seek(d) = \begin{cases} 1.867 * 10^{-3} + 1.315 * 10^{-4} \sqrt{d} & ; d < 1344 \\ 3.8635 * 10^{-3} + 2.1 * 10^{-6}\, d & ; d \ge 1344 \end{cases}$		
mean of fragment size	*E[S]*	*200 KBytes*
variance of fragment size	*Var[S]*	*(100 KBytes)²*
round length	*t*	*1 s*
number of rounds	*M*	*1200*
tolerated number of glitches per stream	*g*	*12*

Table 1: Disk and Data Characteristics of the Simulation

Figure 1 shows the analytically predicted and the simulated values for p_{late}, the probability for suffering one or more glitches in one round, as a function of the multiprogramming level N for a round length t of 1 second. As the chart shows, the analytic model is conservative in that it always overestimates p_{late}, but, on the other hand, the analytic predictions are sufficiently accurate to be usable for driving the admission control of a multimedia server. For example, suppose that the application can tolerate a lateness probability of say 1 percent. According to the analytic model, the multiprogramming level N would have to be constrained to be at most 26, whereas the simulations show that the system could actually sustain 28 concurrent streams. This minor suboptimality seems to be an affordable price for being able to give mathematically derived stochastic service guarantees rather than relying on pure trial-and-error experimentation.

For the case where the quality of service constraint is given by the percentage of glitches that are allowed to occur during the playback duration of $M = 1200$ rounds, our simulations show that a multiprogramming level N of 31 concurrent streams is possible for a tolerable glitch rate of $g/M = 0.01$ and a threshold $\epsilon = 0.01$ for p_{error}.

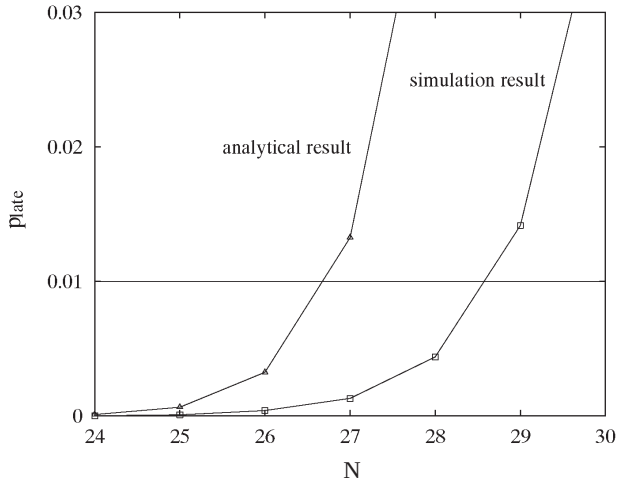

Figure 1: Analytically Predicted vs. Simulated p_{late} Probability

The analytic bound according to (3.3.6) would be 28 concurrent streams. Numerical results for this scenario are given in Table 2.

number of streams N	*analytic results* p_{error}	*simulation results* p_{error}
28	0.00014	0
29	0.318	0
30	1	0
31	1	0.00678
32	1	0.454

Table 2: Analytic Versus Simulation Results for p_{error}

Comparing this result to a deterministic worst-case calculation nicely demonstrates the substantial benefit of our stochastic approach. Namely, a worst-case strategy would limit the number of concurrent streams N by the following equation

$$N_{\max}^{wc} = \left\lfloor \frac{t}{T_{rot}^{\max} + T_{seek}^{\max} + T_{trans}^{\max}} \right\rfloor \qquad (4.1)$$

For the values t = 1s, $T_{rot}^{\max}$ = 8.34ms, $T_{seek}^{\max}$ = 18ms, and $T_{trans}^{\max}$ = 71.7ms, which corresponds to the 99-percentile of the Gamma distribution of the fragment sizes and a disk transfer rate of C_{min} / ROT, we obtain $N_{\max}^{wc}$ = 10. Even optimistically assuming a disk transfer rate of (C_{max} + C_{min}) / (2∗ROT) and lowering the request size, for example, to the 95-percentile of the fragment size distribution, would not result in significantly better estimations. In this case, $T_{trans}^{\max}$ would be 41.9ms, and the number of concurrent streams would be limited to $N_{\max}^{wc}$ = 14.

5 System Issues

The aim of the previous sections was to derive and validate stochastic guarantees for an upper bound of the glitch rate of streams. In this section, we discuss how to utilize these results in the implementation of a multimedia server.

Given a fixed system configuration (i.e., the number and characteristics of the disks) and fixed data characteristics (i.e., the fragment size distribution), the glitch rate of a stream depends only on the total number of streams, N, that are concurrently served. Thus, the glitch rate is bounded with high probability (as derived in Section 3) by allowing only a limited number, N_{max}, of concurrent streams. To implement this form of admission control, we suggest using a lookup table with precomputed values of N_{max} for different tolerance thresholds of the glitch rate. This scheme incurs almost no run-time overhead. The table has to be updated by re-evaluating the analytic model only if the disk configuration or general data characteristics change.

We are currently building a prototype of a multimedia server as part of the Esprit long-term research project HERMES [Her96]. The prototype is able to handle multimedia data of different types, with different bandwidth requirements, and also variable bandwidth within an object, as required by MPEG-2. The architecture of our server in terms of data placement and load balancing is based on the FIVE prototype [SWZ94, SWZ96], an experimental file system for parallel disk systems. We are currently extending FIVE to support the presented admission control.

6 Future Work

Except for requiring a minimal buffer on the client site for incoming fragments, we have disregarded buffering issues so far. In the advanced multimedia applications that we are aiming at, many clients are quite powerful PCs or workstations that have memory and also local disk resources available. Buffering data on the server and/or the client would enable a more efficient disk scheduling by preloading fragments ahead of time and saving resources for heavy-load periods later. This is an issue for further research.

We assume that advanced multimedia applications such as digital libraries or teleteaching will exhibit a *mixed workload* with massive access to continuous data as well as to conventional, "discrete" data such as HTML text documents and images. We advocate sharing disks between continuous and discrete data, as this provides a much better resource utilization from both a disk space and a disk bandwidth point of view. [NMW97] has investigated a first approach to the analytic modeling of such mixed-workload multimedia servers, but its model is not sufficiently accurate in several regards. We plan to refine this line of models towards multimedia information systems with accurately predictable performance and quality of service.

References

[All90] Arnold O. Allen, *Probability, Statistics and Queueing Theory with Computer Science Applications*, 2nd edition, Academic Press, 1990.

[BGM94] Steven Berson, Shahram Ghandeharizadeh, Richard Muntz, *Staggered Striping in Multimedia Information Systems.* Proceedings ACM SIGMOD Conference 1994, International Conference on Management of Data, Minneapolis, Minnesota, pp.79-90, May 1994.

[Bir95] Yitzhak Birk, *Track-Pairing: a Novel Data Layout for VOD Servers with Multi-Zone-Recording Disks*, IEEE International Conference on Multimedia Computing and Systems (ICMCS'95), Washington D.C., May 1995.

[CBR95] Ariel Cohen, Walter A.Burkhard, P. Venkat Rangan, *Pipelined Disk Arrays for Digital Movie Retrieval*, Proceedings of the International Conference on Multimedia Computing and Systems (ICMCS'95), Washington D.C., May 1995.

[CL96] Huang-Jen Chen, Thomas D. C. Little, *Storage Allocation Policies for Time-Dependent Multimedia Data*, IEEE Transactions on Knowledge and Data Engineering, (8)5, pp. 855-864, October 1996.

[CKY93] Mon-Song Chen, Dilip D. Kandlur, Philip S. Yu, *Optimization of the Grouped Sweeping Scheduling (GSS) with Heterogenous Multimedia Streams*, Proceedings of the ACM International Conference on Multimedia (ACM Multimedia '93), Anaheim, CA, 1993.

[CZ94] Ed Chang, Avideh Zakhor, *Variable Bit Rate MPEG Video Storage on Parallel Disk Arrays*, Proceedings of SPIE Conference on Visual Communication and Image Processing, Chicago, Illinois, pp. 47-60, September 1994.

[CZ96] Ed Chang, Avideh Zakhor, *Cost Analyses for VBR Video Servers*, Proceedings of IS&T/SPIE International Symposium on Electronic Imaging: Science and Technology, San Jose, California, January 1996.

[Chu96] Soon.M. Chung (Editor), *Multimedia Information Storage and Management*, Kluwer, 1996.

[Fel71] William Feller, *An Introduction to Probability Theory and Its Applications, Volume 2*, John Wiley & Sons, 1971.

[GHB94] D. James Gemmel, Jiawei Han, Richard Beaton, Stavros Christodoulakis, *Delay-Sensitive Multimedia on Disks*, IEEE Multimedia, pp. 57-67, 1995.

[GK95] Shahram Ghandeharizadeh, Seon Ho Kim, *Striping in Multi-disk Video Servers*, Proceedings of the SPIE High-Density Data Recording and Retrieval Technologies Conference, October 1995.

[GVK95] D. James Gemmel, Harrick M. Vin, Dilip D. Kandlur, P. Venkat Rangan, Lawrence A. Rowe, *Multimedia Storage Servers: A Tutorial*, IEEE Computer, pp. 40-49, May 1995.

[GKS96] Sharam Ghandeharizadeh, Seon H. Kim, Cyrus Shahabi, Roger Zimmermann, Placement of Continuous Media in Multi-zone Disks, in: S. Chung (Editor), Multimedia Information Storage and Management, Kluwer, 1996.

[Her96] Technical Reports of the ESPRIT Long Term Research Project No 9141, HERMES (Foundations of High Performance Multimedia Information Management), http://www.ced.tuc.gr/hermes.

[HR89] Torben Hagerup, Christiane Rüb, *A Guided Tour of Chernoff Bounds*, Information Processing Letters 33, pp. 305-308, 1989.

[Kle75] Leonard Kleinrock, *Queueing Systems*. *Volume 1: Theory*, Wiley, 1975.

[KH95] Marwan Krunz, Herman Hughes, *A Traffic Model for MPEG-Coded VBR Streams*, Proceedings ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, Ottawa,, Canada, May 1995.

[Nel95] Randolph Nelson, *Probability, Stochastic Processes, and Queueing Theory : The Mathematics of Computer Performance Modeling*, Springer, 1995.

[NMW97] Guido Nerjes, Peter Muth, Gerhard Weikum, *Stochastic Performance Guarantees for Mixed Workloads in a Multimedia Information System*, Proceedings IEEE International Workshop on Research Issues in Data Engineering (RIDE'97), Birmingham, UK, April 1997.

[ÖRS96] Banu Özden, Rajeev Rastogi, Avi Silberschatz, *Disk Striping in Video Server Environments*, Proceedings IEEE International Conference on Multimedia Computing and Systems (ICMCS'96), Hiroshima, Japan, June 1996.

[Oya95] Yen-Jen Oyang, *A tight upper bound of the lumped disk seek time for the Scan disk scheduling policy*, Information Processing Letters 54, pp. 355-358, 1995.

[Ros95] Oliver Rose, *Statistical Properties of MPEG Video Traffic and Their Impact on Traffic Modeling in ATM Systems*, Technical Report 101, Institute of Computer Science, University of Würzburg, Germany, 1995.

[RW94] Chris Ruemmler, John Wilkes, *An Introduction to Disk Modelling*, IEEE Computer, 27(3), pp. 17-28, March 1994.

[SG94] Abraham Silberschatz, Peter Galvin, *Operating System Concepts*.4th edition. Addison-Wesley, New York, 1994.

[SWZ94] Peter Scheuermann, Gerhard Weikum, Peter Zabback, *Disk Cooling in Parallel Disk Systems*, IEEE Data Engineering Bulletin Vol.17 No.3, pp. 29-40, September 1994.

[SWZ96] Peter Scheuermann, Gerhard Weikum, Peter Zabback, *Data Partitioning and Load Balancing in Parallel Disk Systems*, Technical Report A/02/96, Department of Computer Science, University of the Saarland, 1996, submitted for publication.

[SJ96] V.S. Subrahmanian, Sushil Jajodia (Editors), *Multimedia Database Systems: Issues and Research Directions*, Springer, 1996.

[TCG96a] Peter Triantafillou, Stavros Christodoulakis, Costas Georgiadis, *A Comprehensive Analytical Performance Model for Disk-Storage Device Technologies*, Hermes Technical Report No 11, MUSIC, Technical University of Crete, Greece, 1996.

[TCG96b] Peter Triantafillou, Stavros Christodoulakis, Costas Georgiadis, *Optimal Data Placement on Disks: A Comprehensive Solution for Different Technologies.*, Hermes Technical Report No 10, MUSIC, Technical University of Crete, Greece, 1996.

[TKKD96] Renu Tewari, Richard P. King, Dilip Kandlur, Daniel Dias, *Placement of Multimedia Blocks on Zoned Disks*, IS&T/SPIE Conference on Multimedia Computing and Networking(MMCN'96), San Jose, California, January 1996.

[TPBG93] Fouad A. Tobagi, Joseph Pang, Randall Baird, Mark Gang, *Streaming RAID - A Disk Array Management System for Video Files*, Proceedings of ACM Multimedia Conference, 1993.

[VGG94] Harrick M. Vin, Pawan Goyal, Alok Goyal, Anshuman Goyal, *A Statistical Admission Control Algorithm for Multimedia Servers*, ACM Multimedia Conference, 1994.

[VGG95] Harrick M. Vin, Alok Goyal, Pawan Goyal, *Algorithms for Designing Large-Scale Multimedia Servers*, Computer Communications, March 1995.

[Won83] C.K. Wong, *Algorithmic Studies in Mass Storage Systems*, Computer Science Press, 1983.