On Mixed-Workload Multimedia Storage Servers with Guaranteed Performance and Service Quality *

G. Nerjes[2], Y. Romboyannakis[3], P. Muth[1],
M. Paterakis[3], P. Triantafillou[3], G. Weikum[1]

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

[2] Swiss Federal Institute of Technology
Institute of Information Systems
CH-8092 Zurich, Switzerland
nerjes@inf.ethz.ch

[3]Technical University of Crete
Department of Electronics &
Computer Engineering, Chania, Greece
{pateraki, peter, rombo}@ced.tuc.gr

Abstract

An important issue in multimedia information systems that has received considerable attention is to provide performance and service quality guarantees for "continuous" streams of video/audio data, especially bounding the rate of non-timely data fragments, so-called "glitches", under a given number of concurrently served streams. An equally important but much harder problem that has been neglected so far is to extend such guarantees to a mixed workload with both continuous-data streams and response-time-sensitive requests to conventional, "discrete" data. This paper develops an analytic performance model for such a mixed workload. The model is a hierarchical one, where the higher, macroscopic level addresses the mutual performance impacts of continuous-data and discrete-data requests by means of an abstract Markov process model, and the lower, microscopic level analyzes the glitch rate under a detailed disk model. A configuration method for mixed-workload multimedia storage servers on the basis of this hierarchical model is finally presented.

1 Problem Statement

Multimedia information servers that manage very large volumes of disk-resident video/audio as well as text and image data have to meet stringent performance requirements. Most notably, the real-time nature of "continuous" data (*C-data*) like video/audio requires that a server can guarantee a sufficiently short response time for each of the data fragments (e.g., all successive video frames corresponding to one second of display) that constitute the continuous data stream between the server and the client, throughout the display duration of the entire C-data object. Otherwise, fragments that arrive too late because of not meeting the response time goal are perceived as "hiccups" during the display. Such timing problems, which we refer to as "glitches" in this paper, can be avoided

*This work has been supported by the ESPRIT Long Term Research Project HERMES

Proceedings of the 3rd Intl. Workshop on Multimedia Information Systems (MIS'97), Como, Italy, September 1997

by analyzing the worst-case response time of a fragment access for a given number of concurrently sustained data streams (e.g., video viewers) and not admitting any additional new streams if the predicted response time under the new increased load exceeds the specified goal. The results of such an analysis can be used also for configuring a server, particularly, determining the necessary number of disks, so that it can definitely sustain an expected number of concurrent streams for a specific application (e.g., a teleteaching or news server).

A number of papers along these lines have appeared in the recent literature (e.g., [ORS96, GKS96, GHBC94, YCK93]). As the workload parameters (e.g., fragment sizes) and also certain parameters on the server side (e.g., disk latencies) can often be characterized only stochastically and a very small number of glitches can usually be tolerated in most multimedia applications, some of these papers have pursued stochastic models in that they consider probabilistic bounds on the server response time (e.g., the 99th percentile) and resulting glitch rate [VGGG94, CZ94, NMW97b]. In this paper we also adopt such a stochastic view of performance and service quality guarantees, as this allows a much better resource utilization and thus offers a much better price/performance ratio than a conservative worst-case model.

The prior work on storage management for multimedia information systems has focused almost exclusively on C-data. However, advanced applications such as teleteaching, digital libraries, or virtual museums incur a mixed workload in that they certainly require access also to conventional "discrete" data (*D-data*) such as text or image documents. In [MNO+96] this is taken into account by reserving a fixed fraction of the server's performance capacity for such D-data requests. However, this is merely a best-effort approach with regard to the response time of those requests. The server should be configured such that it can also give stochastic guarantees with regard to the response time of D-data requests (e.g., its 90th percentile). Further note that both C-data and D-data should reside on the same shared disk pool for cost/performance reasons, so that load fluctuations of the two request classes can be balanced out across both classes.

The rest of the paper is organized as follows. Section 2 describes our architectural model and assumptions. Section 3 develops the two building blocks of the hierarchical model, the macroscopic and the microscopic models. Section 4 presents the configuration algorithm, which is based on the two stages of the hierarchical model, and section 5 describes some implementation issues and gives an outlook on future work.

2 Our Approach

Our observation is that performance and service quality guarantees for mixed workloads is a totally neglected area within multimedia information systems that will be of crucial importance for the acceptance of future multimedia client-server applications. Therefore, we have started working

towards an analytic understanding and a configuration methodology for mixed-workload servers with stochastic guarantees for both C-data and D-data requests. The only prior work that has tackled this problem is [NMW97a] that uses a simple form of M/G/1 vacation server model [Tak91] for reconciling C-data and D-data requests. However, that paper eventually had to resort to simulation-based modeling because of the inaccuracy of its underlying analytic model while more elaborated variants of M/G/1 vacation server models have not been considered for tractability reasons.
We have developed a tractable and sufficiently accurate analytic model for the stochastic characterization of the performance and service quality of a mixed-workload server. The inherent mathematical complexity of the problem setting is reduced by dividing the overall problem into two subproblems and addressing them in a hierarchical model. At the higher level of the model, we take a "macroscopic" view of mixed workloads and focus on the mutual impact of C-data and D-data requests while disregarding the details of the underlying disk service and rather studying a high-level Markov process model. In particular, an entire C-data stream is viewed as one request and there is no notion of glitches at this level. At the lower, "microscopic" level of the model, these omissions are corrected by analyzing in detail the glitch rate per C-data stream under a realistic disk service model while ignoring the impact of D-data requests at this stage. The two levels of the hierarchical model are used in the following way:

1. The macroscopic model allows us to derive the required amount of abstract service capacity (which will finally be translated into the required number of disks) to meet specified guarantees with regard to the tail of the probability distribution of

   a. the waiting time $P(\omega_D > t)$ of D-data requests (i.e., their response time minus the service time which is spent in any case) and

   b. the start-up delay $P(\omega_C > t)$ of a new C-data stream

   under the assumption of Poisson-process arrivals [All90, Nel95] for both C-data streams and D-data requests. An important measure that is computed here is the probability distribution for the number of concurrent C-data streams $P(N_C)$.

2. The microscopic model allows us to derive the required number of disks D to meet specified service guarantees in terms of the glitch rate per C-data stream, for a given number of concurrent streams. Its input is the probability distribution for the stream population $P(N_C)$ as computed by the macroscopic model, and it effectively translates the abstract notion of service capacity into a number of disks D.

This hierarchical analytic model can be used for a) predicting the performance and service quality guarantees that a server with a given configuration can provide, and b) for determining the required

configuration so that specified guarantees can be satisfied. The focus of the paper is on the second goal. So, in addition to developing the analytic model itself, we present a heuristic method for configuring a mixed-workload multimedia storage server. We are not aware of any prior solutions to this challenging and practically very important problem.

3 Overview of the Model

The model consists of two levels. The first level takes a macroscopic view of the system in which there are two classes of requests submitted to the multimedia server by its clients: the continuous stream requests, and the discrete data requests. As a result, the overall system consists of two subsystems, one serving the continuous stream requests, the C-system, and another serving the discrete data requests, the D-system. C-system customers at this level represent high-level requests like the playback of an entire video. At this level of the model we study issues pertaining to the interdependencies between the performance experienced by C-system and D-system customers and quality of service (QoS) metrics that affect these high-level requests (such as start-up latency). The second level of the model takes a microscopic view of the C-system. C-system customers at this level represent low-level requests such as the individual I/O requests issued during the playback of a continuous data object. This level allows stochastic evaluation of the QoS metrics relevant to this low level, especially glitch rates. The overall model is based on the following two-class service discipline: (i) a certain fraction of the overall service capacity is reserved for D-system customers and the remaining capacity is given to the C-system, and (ii) unused capacity of the C-system can be dynamically transferred on demand to the D-system from where it can be reclaimed back by the C-system upon the departures of D-system customers (i.e., without preemption).

3.1 The Macroscopic View

3.1.1 C-System

The C-system is modeled as an $M/M/n_C/k_C/FCFS$ queue with n_C servers, a finite queue capacity k_C, and a first-come-first-served (FCFS) service discipline. The service rate for each of the n_C servers is denoted by μ_C. The arrivals of C-stream requests to the server are assumed to be Poisson with arrival rate λ_C. Furthermore, the duration of a C-stream is assumed to be exponentially distributed with mean d_C ($d_C = 1/\mu_C$). The tail of the probability distribution of the startup delay $P(\omega_C > t)$ of a new C-data stream is then given by

$$P(\omega_C > t) = \sum_{i=n_C}^{k_C-1} \pi_i \sum_{m=0}^{i-n_C} e^{-n_C\mu_C t} \frac{[n_C\mu_C t]^m}{m!} \tag{1}$$

where π_i denotes the probability that there are i C-stream requests in the system. Its value can be obtained solving the following equations.

$$\pi_i = \begin{cases} \pi_0 \frac{(\frac{\lambda_c}{\mu_c})^i}{i!} & , \quad 1 \leq i \leq n_C \\ \pi_0 (\frac{\lambda_c}{\mu_c})^i * \frac{n_C^{(n_C - i)}}{n_C!} & , \quad n_C < i \leq k_C \end{cases} \qquad \text{and} \quad \sum_{i=0}^{k_C} \pi_i = 1 \tag{2}$$

3.1.2 D-System

The D-system is modeled by an $M/M/1/k_D/FCFS$ queue, with Poisson arrivals of discrete data requests with rate λ_D, a finite queue capacity of k_D requests, and an FCFS service discipline. The overall service rate of the D-system varies depending on the occupancy of the C-system, which in essence implies that the number of servers n_D is a *variable* whose value depends on the occupancy of the C-system. The minimum value of n_D, denoted by n_D^{min}, is such that $n_D^{min} \cdot \mu_D = \mu_D^{min}$, where μ_D, denotes the service rate of each server ($1/\mu_D$ is equal to the average service demand of a discrete data request), and μ_D^{min} denotes the minimum aggregate service rate dedicated to the D-system. The variable overall service rate $\mu_D^{var}(j)$ is given by

$$\mu_D^{var}(j) = \mu_D^{min} + \mu_D * \max(0, n_C - j) \tag{3}$$

where j denotes the number of continuous streams present in the system.
The probability distribution of the variable number of D-system servers is analyzed by means of a 2-dimensional Markov chain whose states are given by the numbers of C-system customers and D-system customers in service or being queued. The balance equations for this Markov chain are solved numerically, truncating the infinite state space to a finite subset such that the "missing" probability is negligible. As a result the tail of the probability distribution of the waiting time $P(\omega_D > t)$ of D-data requests can be computed by

$$P(\omega_D > t) = \sum_{i=1}^{k_D - 1} \sum_{j=0}^{k_C} \pi_{i,j} \sum_{m=0}^{i-1} e^{-\mu_D^{var}(j)t} \frac{[\mu_D^{var}(j)t]^m}{m!} \tag{4}$$

$\pi_{i,j} = P[N_D = i \wedge N_C = j]$ denotes the probability of having i D-system and j C-system customers at the same time.

3.2 The Microscopic View

At the microscopic level of the model the disk performance is explicitly modeled for stochastically evaluating the per stream "glitch rate", defined as the probability that a given data fragment of a C-data object is not fetched from disk and delivered to the client before its display deadline. This level also takes into account the variability of the display bandwidth both across different C-data objects and within an object, and the resulting variable size of data fragments. In [NMW97b] we have derived

an accurate stochastic model that is briefly summarized here. This model analyzes the probability distribution of the total disk service time for serving a given number of data fragment accesses on the same disk. A periodic (round-based) scheduling discipline and constant-time-length data fragments of duration T are assumed. The fragments are spread across the disks in a coarse-grained striping layout such that each fragment resides entirely on one disk [ORS96, BGMJ94, TPBG93]. This allows us to consider only a single disk for the rest of this section as there are no scheduling dependencies between different disks. In particular, the model derives the Laplace-Stieltjes transform of the total service time, using a tight upper bound for the total seek time of the scan disk-arm sweep and considering the impact of variable fragment sizes.

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

$$T_N = T_{seek} + \sum_{i=1}^{N} T_{rot,i} + \sum_{i=1}^{N} T_{trans,i} \tag{5}$$

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. Using the distribution function of these random variables, i.e. uniformly distributed rotational delays and Gamma distributed fragment sizes we obtain the Laplace-Stieltjes transform T_N^* of T_N. Now Chernoff's theorem [Kle75, Nel95] allows us to bound the tail probability of the random variable T_N by:

$$P[T_N \geq T] \leq \inf_{\theta \geq 0} \left\{ e^{-\theta T} \; T_N^*(-\theta) \right\} = b_{late}(N, T) \tag{6}$$

In [NMW97b] it is shown how $b_{late}(N)$ can be used to derive a upper bound of the tail probability that a stream with a duration of M rounds suffers more than a given percentage r of glitches

$$P\left[\begin{array}{l} \text{number of glitches in} \\ \text{a stream in } M \text{ rounds} \end{array} \geq r * M \right] \leq b_{error}(N, T, M, M * r) \tag{7}$$

This bound depends on the number of rounds M, the number of streams N and the duration T of a scheduling round. To derive the probability for the overall glitch rate we have to take the duration of streams into account. In section 3.1.1, we have assumed the duration of streams to be exponentially distributed with mean d_C. Now the duration of a stream has to be translated into a discrete number of R rounds. The probability for a stream to have M rounds is:

$$P[R = M] = P[(M-1) * T \leq d \leq M * T] = e^{-\frac{1}{d_C}(M-1)*T} - e^{-\frac{1}{d_C}M*T} \tag{8}$$

We are now ready to give a bound on the total probability for a stream to have a glitch rate higher than a given percentage r. Because of the binomial distribution of the probability that a stream suffers a given number of glitches over its number of rounds the probability for a stream to exceed a certain glitch rate decreases with its size. Therefore, we can conservatively approximate

the probability that the overall glitch rate becomes higher than r by assuming the glitch rate for all streams longer than M_{max} rounds to equal the glitch rate of a stream of M_{max} rounds:

$$\begin{aligned} p_{grate}(N) &= P[\text{glitchrate} \geq r] \\ &= \sum_{M=0}^{\infty} P[R=M] * P\left[\begin{array}{c} \text{number of glitches of} \\ \text{a stream in } M \text{ rounds} \end{array} \geq M * r \right] \\ &\leq \sum_{M=0}^{M_{max}} \left(\left(e^{-\frac{1}{d_C}(M-1)*T} - e^{-\frac{1}{d_C}M*T} \right) b_{error}(N,T,M,M*r) \right) \\ &\quad + e^{-\frac{1}{d_C}M_{max}*T}\, b_{error}(N,T,M_{max},M_{max}*r) \end{aligned} \tag{9}$$

The value of M_{max} should be chosen such that most streams have less rounds than M_{max}, for example by choosing M_{max} such that it equals the 99th percentile of the distribution of the number of rounds.

4 Towards a Configuration Tool for Mixed-Workload Servers

So far we have developed the two building blocks of our two-stage analytic model. In this section we show how both blocks, the macroscopic and the microscopic block, are coupled and how we can construct a configuration tool for mixed-workload servers.

We can assess all relevant performance and quality of service metrics in a coherent manner by specifying bounds for C-data startup delay, D-data waiting time and C-data glitch rate, and derive, via binary search over relatively small value ranges, the required numbers of abstract servers, n_C and n_D, and ultimately the required number of disks, D. This consideration leads to the following procedure:

1. Iterate on the macro model, with different values of n_C, until the obtained bounds on the C-stream start-up delay are acceptable. This step should also determine the state probabilities $\pi_i = P[N_C = i]$, where N_C denotes the number of customers in the C-system.

2. Iterate on the micro model until the minimum number of disks D is obtained that guarantees the specified glitch rate. As the micro model is conditional on the population of C-system customers, i.e., the number of concurrently served C-data streams, the evaluation for a given D must be repeated for all possible values of N_C and weighted by the corresponding marginal state probabilities $P[N_C = i]$ to obtain the total probability that a certain glitch rate is not exceeded. If the specified guarantee is not met, D is increased until the desired bound on the glitch rate is achieved. Finally D can be converted into a number of servers n_C.

3. Set the number of servers n_C to the maximum number derived from steps 1) and 2).

4. With maximum n_C the D-system part of the macromodel is evaluated to obtain the minimum value of n_D to ensure that the response time goal for D-requests is met.

5. The final coupling of the two model stages is based on relating their corresponding notions of servers. At the macro level, a server represents the service capacity to serve exactly one C-data stream of average display bandwidth or exactly one D-data request at a time, respectively, where the service demand of the D-data requests is derived from the average size of a D-data object. From the average C-data bandwidth and the average D-data size we can infer the average disk service time that is needed for one round of a C-data stream and one D-data request, respectively, using a straightforward disk model that incorporates average seek time, rotational latency, and transfer time. We denote these average disk service times by $1/\mu_{disk,C}$ and $1/\mu_{disk,D}$. Then the number of disks, D, that corresponds to n_C and n_D servers at the macro level is given by:

$$D = n_C * \frac{1}{\mu_{disk,C}} + n_D * \frac{1}{\mu_{disk,D}} \tag{10}$$

The overall configuration algorithm is outlined in pseudo-code in Figure 1.
This form of coupling the two stages of the hierarchical model may appear overly simple at a first glance, as it is solely based on average service demands and disregards the variability that is crucial in terms of the response time distribution of the D-system and the glitch rate of the C-system. However, these two critical metrics are exactly taken care of by the macro and micro models themselves, so that the coupling can indeed be simplified.

5 Concluding Remarks

The analytic model forms the core of a configuration tool for mixed-workload multimedia information servers that we are aiming to develop. The configuration algorithm outline in the previous section has been implemented in C with approximately 400 lines of code, using the Gauss-Jordan elimination algorithm for solving the two-dimensional Markov chain. Initial tests have shown that the evaluation of the model is very efficient. Validation studies for comparing the model's results to a detailed simulation are underway.
Future work will include reconsidering some of the modeling assumptions in our approach and possibly generalizing the model. In particular, the assumption that the disk service time of D-requests is exponentially distributed is primarily due to tractability reasons. Considering general distributions of the D-request service time, for example, by analyzing the embedded Markov chain at the request departure timepoints, would certainly further improve the model's accuracy. In addition, it would be desirable to incorporate more details of the actual disk scheduling policy into the model,

0) Compute service rates of actual disks:

$\mu_{disk,C} := \frac{\text{disk bandwidth}}{\text{display bandwidth}}$ $\mu_{disk,D} :=$ disk service rate of D-requests based on the mean request size and mean disk latency

1) Iterate the $M/M/n_C/k_C/FCFS$ part of the macro model to determine the minimum value of n_C to ensure the startup delay goals for C-data streams (this step also determines the state probabilities $P[N_C = i]$

2) Iterate the micro model to determine the minimum number of disks, D, necessary to ensure the glitch rate goal for C-data streams

   2.1) For i = all N_C values:
   evaluate the micro model to determine $p_{grate}(\lceil \frac{i}{D} \rceil)$

   2.2) Compute the overall glitch rate probability $p_{grate}^{total}(D)$ as the total probability

$$p_{grate}^{total}(D) = \sum_{\text{i = all } N_C \text{ values}} p_{grate}(\lceil \frac{i}{D} \rceil) * P[N_C = i]$$

   2.3) set $n_C := D * \mu_{disk,C}$

3) Set n_C to the maximum of the two values obtained from steps 1 and 2

4) With fixed n_C, iterate the twodimensional Markov chain part of the macro model to determine the minimum value of n_D to ensure the waiting time or response time goal for D-requests

5) Set the total number of required disks to $\frac{n_C}{\mu_{disk,C}} + \frac{n_D}{\mu_{disk,D}}$

Figure 1: Pseudo-Code Algorithm for Server Configuration

and to be able to study a larger variety of scheduling policies (e.g., more refined variants of a SCAN algorithm that may treat C-requests and D-requests in a non-uniform way).

References

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

[BGMJ94] 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.

[CZ94] 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.

[GHBC94] D. James Gemmell, Jiawei Han, Richard J. Beaton and Stavros Christodoulakis. Delay-Sensitive Multimedia on Disks. *IEEE Multimedia*, pages 57–67, 1994.

[GKS96] Shahram Ghandeharizadeh, Seon Ho Kim and Cyrus Shahabi. On Disk Scheduling and Data Placement for Video Servers. *ACM Multimedia Systems*, 1996.

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

[MNO+96] 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.

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

[NMW97a] 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*, April 1997.

[NMW97b] 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*, May 1997.

[ORS96] 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.

[Tak91] Hideaki Takagi. *Queueing Analysis : A Foundation of Performance Analysis, Volume 1 : Vacation and Priority Systems.* North-Holland, Amsterdam, 1991.

[TPBG93] 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.

[VGGG94] 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.

[YCK93] 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.