Admission Control for Media on Demand Services

Martin Bichler, Thomas Setzer

Roland Berger and 0_2 Germany
Chair of Internet-based Information Systems (IBIS)
Institute of Informatics, Technische Universität München
Boltzmannstraße 3, 85748 Garching, Germany
Email: bichler@in.tum.de, setzer@in.tum.de
Phone: ++49-89-289-17500

Abstract

Admission control software is used to make accept or deny decisions about incoming service requests to avoid overload. Existing media streaming software includes only limited support for admission control by allowing for predefined static rules. Such rules limit for example the number of requests that are allowed to enter the system during a certain time or define thresholds concerning the utilization level of a single resource such as network bandwidth. In media streaming applications, however, the bottleneck resource (CPU, Disk I/O, network bandwidth, etc.) might change over time depending on the current demand for different types of audio or video files. This paper proposes a model for adaptive admission control in the presence of multiple scarce resources. Opportunity costs for a service request are determined at the moment of an incoming request and compared to the revenue of a request in order to make an accept/deny decision. Opportunity costs are based on resource utilization, service resource requirements, expected future demand for services, and the revenue per accepted service. The model allows rejection of service requests early to reserve capacity required to perform future service requests with higher revenues. We describe a number of experiments to illustrate the benefits of adaptive admission control models over static admission control rules.

Key words: *Admission Control, IT Service Management, Media Streaming, Service Level Management*

Introduction

Service-oriented architecture (SOA) expresses a perspective of software architecture that defines the use of loosely coupled software services to support the requirements of the business processes and software users. With increased

penetration of broadband networks multimedia services are becoming an integral part of service-oriented architectures. Multimedia services are nowadays used as part of enterprise and educational services as well as for professional Video on Demand services. According to market research companies such as Frost&Sullivan [1] technological developments such as higher compression rates are going to lead to further growth as the viewing experience for end-users using streaming services gets better.

Media Streaming

There are two ways in which media content can be delivered over the Internet - using a Web server (Download-and-Play) or by using a media streaming server. The Download-and-Play model requires clients to first download the whole media file and then run it on their desktops. Depending on bandwidth, e.g. a 90-minute MPEG4 encoded movie could take approximately ninety minutes to download on a 1-Mbps Internet connection. This means that the time taken by the viewer to download the media is as long as the time taken to view it. Furthermore, a content provider does not have control over who is potentially redistributing the file. While Web servers work well for static media such as text and images, they have the disadvantages mentioned when serving multimedia data like video and audio files to viewers on demand. ”On demand” means that customer get instant access to a media streaming services and pay only for the individual service request.

When delivering media via a media streaming server (media server), a client does not have to download the full content before playing it. When using a media streaming server, a media file is actively streamed to the client at the exact data rate associated with the compressed audio and video streams that is needed by a client application to play the file without discontinuation. Typically, one would face just a few seconds time lag for buffering information before a video gets displayed on the client. This small buffer allows the media to continue playing uninterrupted even when there is network congestion or other kinds of system overload. Our work focuses on media streaming servers.

Media Streaming Server Overload

A media streaming infrastructure is sized and designed to deliver continuous media streams to clients [2, 3]. However, system malfunctions, unexpected service behavior or peak demands for one or more media services may lead to overload and one or several critical resources such as network interface, disk, physical memory or CPU become scarce. During such times of high load, the demand exceeds the capacity of the infrastructure and there are not sufficient resources to provide adequate quality for streaming media to all clients.

The IT Infrastructure Library (ITIL), one of the most widely used guidelines for IT service management, is a framework of best practices intended to achieve high quality and value for money in IT operations [4, 5]. According to ITIL, service level management (SLM), as one of the major tasks of IT Service Management (ITSM), is responsible for the provisioning of IT Services according to quality attributes arranged in service level agreements. A central aspect herein is overload control (OC), which addresses the handling of overload or rather the prevention of overload situations. Due to the real-time requirements in media streaming, efficient overload control is vital. An overloaded media streaming platform cannot keep up with the delivery of data packets to guarantee a smooth, seamless play of the media, producing jerky video or audio [6] [3] [7].

Admission Control to Handle Overload

Admission control mechanisms are making decisions about accepting, buffering or rejecting incoming service requests to avoid overload [8, 9]. Existing media infrastructure design includes only limited support for admission control by allowing only for predefined static rules. Such rules limit for example the number of requests that are allowed to enter the system during a certain time or define thresholds concerning the utilization level of a single resource such as network bandwidth or CPU [10].

Service Differentiation

Most existing admission control algorithms for media streaming services treat every connection request equally [11]. Professional service providers differentiate

prices of their products or they differentiate by customer. For example, a Video on Demand (VoD) service provider may offer different movies and perhaps different quality (HDTV vs. MPEG 4) at different prices. Additionally, a VoD may offer guaranteed service to premium customers, paying a monthly base-fee and a best effort service to standard customers. In order to maximize revenue and better fulfill service level agreements, prioritization of service requests is a useful strategy.

Adaptivity in Admission Control

The evidence from media workload analysis indicates that client demands are highly variable [12-14]. Implementing admission control by setting static rules, like for example “accept only incoming requests for standard services if less then 150 streaming clients are currently connected” defined a priori works well only in steady workload situations [15]. There are a number of problems with static admission control rules: If one chooses low thresholds, server resources may not be fully utilized causing loss of revenue because lower prioritized service request might get denied although enough resources are available. If one chooses high thresholds, it is possible to achieve higher utilization and throughput for low-priority service requests, but there is a risk of overload and high response times if the demand for prioritized services is higher than expected. In order to facilitate optimal admission control, resource utilization and workload demand need to be taken into account to dynamically adapt admission control policies.

Multiple Scarce Resources

High quality streaming to a large number of clients imposes significant demands on different server resources [16], [17]. By using stress tests and bottleneck analyses on media servers, Cherkasova et al. [2, 18] among others, found that depending on the current demand for services some server resources can be over utilized, while the demand on other resources is low because certain types of media streams utilize one resource (bandwidth, CPU, hard disk access, memory etc.) more than others. As a consequence, the bottleneck resource can change over time depending on the demand mix and admission control taking into account only a single resource is suboptimal.

In this paper, we propose a method considering service differentiation in the presence of multiple scarce resources. The rest of the paper is structured as follows: In the next section an overview is given of existing admission control techniques for media streaming services. Subsequently, our admission control decision model formulation is described. Finally, the experimental setup that has been used and the experimental results are described.

State-of-the-Practice and Related Literature

In widespread media streaming server products that are currently on the market (such as Apple's QuickTime Streaming Server, Real Networks Helix Streaming Server, Microsoft Streaming Services, Macromedia Flash Video Streaming Service [19] [20] [21] [22]) it is possible to set threshold values for a maximum *number of allowed parallel streaming connections*, and for the *maximum overall streaming bandwidth or throughput*. Sometimes, it is possible to set these values also for collections of files, which allows some level of static differentiation. In addition, limits in capacity usage of main memory and cache can sometimes be set to limit capacity usage of the server software in total.

Most admission control approaches for media streaming services found in literature do not consider multiple different service classes. Their main goal is to accept as many clients as possible without violating overall QoS requirements. To achieve this, they allow predefined static rules associated with the number of incoming or in-process connections and bandwidth allocation are used as indicator of high load [10]. Static admission control policies for multimedia infrastructures to allow higher utilization levels under high statistical assurances to keep Service Level Agreements have for example been analyzed by Vin et al. and by Kwon and Yeom [9, 23, 24].

Differentiating admission control mechanisms for media streaming services have been investigated by Chen et al. [10]. The authors cluster services into different priority classes. Requests belonging to a certain service class are successively denied depending on certain current utilization levels of a particular resource. Their algorithm is adaptive in sense that it is driven not only by hardware

requirements, but also by analyzing the workload characteristics and trends of client requests, thus allowing the system to adjust dynamically in response to changes in client workload characteristics. Multiple bottleneck resources are not considered.

Welsh et al. analyzed a generic admission control architecture called Staged Event-Driven Architecture (SEDA) to handle multiple bottlenecks in IT infrastructures [25-27]. The idea of SEDA is to model server resources as a network with multiple stages connected with explicit event queues coupled with admission control to prevent resource overload. No special admission control strategy is proposed and each stage may implement its own associated admission control strategy. Thus, strategies allowing for service differentiation might be implemented at each stage. As each stage represents a certain server, this approach allows for overload control in the presence of multiple bottleneck resources only if the bottleneck resources are independent in a sense that they are associated with different physical servers. Furthermore, the problem of allocating multiple scarce resources to differentiated services efficiently is not addressed in this work. In this paper we focus on this specific resource allocation problem.

Decision Model

We will first introduce a basic admission control model for shared IT infrastructures named DLP, underlying a number of restrictive assumptions. Based on the basic model formulation, a couple of extensions and heuristics are introduced to consider relevant conditions you find in practice like time continuous service demand and stochastic resource utilization.

Basic Model and Shadow Prices

A Media on Demand service provider offers services of I different service classes i ($i = 1, \ldots, I$), requested stochastically in discrete points in time t_k ($k = 0, \ldots, \infty$). Service demand D_i is a positive random variable for which we assume some discrete (e.g., a Poisson) distribution. r_i is the revenue of a service request for service class i. The service duration, i.e. the length of time a requested service uses resources, is of fixed length Δt ($\Delta t = t_{k+1} - t_k$). Thus, a service is finished right

before the next possible request time t_{k+1}. Resource allocation coefficients a_{ei} represent the capacity a resource $e = 1, \ldots, E$ is required during Δt by a service i. A resource e has a fixed, limited capacity C_e (see Figure 1).

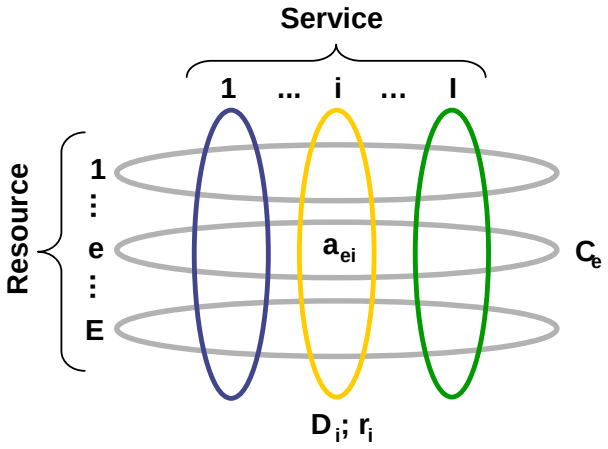

Figure 1 Resource Allocation Matrix

The task is to allocate available resource capacity units to incoming service requests in a way that overall profit is maximized. Based on these assumptions, the problem can be modeled as the following Integer Program (IP):

$$\begin{aligned} \max \quad & \sum_{i \le I} r_i \cdot x_i \\ s.t. \quad & \sum_{i \le I} a_{ei} x_i \le C_e \qquad \forall e \le E \\ & x_i \le D_i \qquad \forall i \le I \\ & x_i \in Z_+ \qquad \forall i \le I \end{aligned} \qquad \text{(IP)}$$

The positive integer variable x_i describes the number of accepted requests for a service of class i in a planning period Δt. In order to avoid the computational complexity of IPs, one can use the LP relaxation of IP and replace the positive random variable D_i by its forecast or expected value, which we will call the Deterministic Linear Program (DLP). Similar linear programs are being used in airline yield management, where we have a large number of bookings [28]. The dual variables λ_e of the capacity constraints of this LP relaxation represent shadow prices or opportunity costs of allocating a unit of resource e. Overall opportunity costs of a service request i can be calculated as the sum of products of resource allocation coefficients and opportunity costs per resource unit $(\sum_e a_{ei}\lambda_e)$ [29]. A

request is accepted, if its revenue exceeds its corresponding overall opportunity costs.

Continuous Demand

The basic model formulation assumes arrivals of service request in discrete points in time and equal length for all service requests. Accordingly, all resource units of all resources are available at the beginning of each planning period, so capacity restrictions in DLP are set to maximum capacities C_e for each resource e. These assumptions are clearly idealized and are only appropriate in special environments, for example in case of batch jobs of equal length.

In practice, Media on Demand service providers are mostly faced with continuous demand as requests might arrive anytime, and services have different resource requirements and different durations. Furthermore, at the beginning of a planning period capacity units might already be allocated to existing media streams. These units are unavailable during the planning period until active streams are processed. The determination of resource workload at the beginning of a planning period provides the available capacity units C_e. Admission control decisions based on these capacity restrictions may results in high shadow prices, as this would imply that resource units currently allocated remain allocated throughout the entire planning period. However, after active services have finished during a planning period, resource units are again available for incoming service requests.

We extend the resource allocation matrix by a time dimension t_{ei} describing the duration for which a service i allocates a_{ei} units of a resource e. Due to varying service demand over time, resources might be scarce in a certain moment, and, after a few service requests have been finished, a larger amount of resource capacity will be available again.

To re-calculate shadow prices, capacity available during a planning period is of interest as well as expectations about the capacity needed by services requests. We developed a number of heuristics to approximate the amount of capacity available

during a planning period and to approximate the amount of expected resource demands during a planning period.

Available Capacity

During the lifetime of a media streaming server, that is, the sequence of all considered planning periods, we have K incoming service requests, each single request, $k=1, \ldots, K$, associated with a certain service class i. At the moment of an incoming request k for a service of class i, t_k, the planning period is set to the estimated duration of the requested service of class i, t_i (see Figure 2), which might be the length of a full movie. For each resource e (e.g., bandwidth, CPU, memory), with $a_{ei} > 0$, the following steps are necessary: The remaining duration $l_{k'}$ of services currently in process are calculated. Allocations before t_k, irrelevant for the current decision, are ignored. For all k' exceeding the planning period duration $(t_k + t_i)$, their remaining allocation durations $l_{k'}$ are limited to the planning interval $l_k=[t_k, t_k + t_i]$ relevant for decision making.

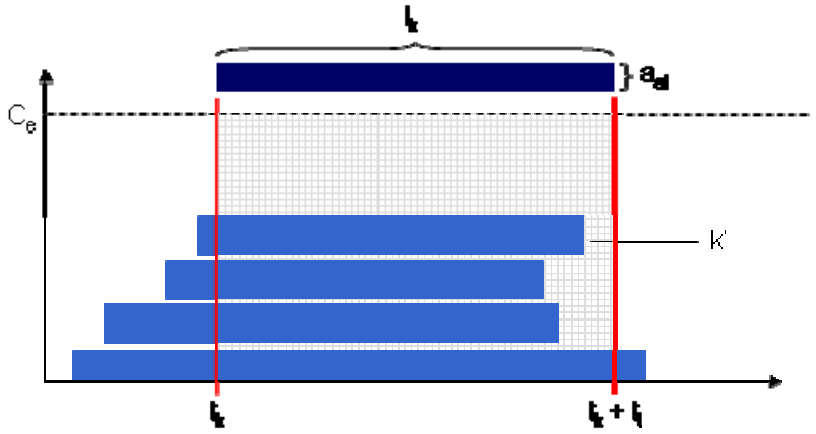

Figure 2 Available Capacity

The sum $\sum_{k'} a_{ei} l_{k'}$ of expected resource allocations of e by active services k' is the amount of resources that is not available for incoming media service requests during the planning period l_k. Subtracting this sum from a resource's maximum capacity in l_k, that is $l_k C_e$ (with C_e as overall capacity of e per time unit) gives an estimate of the amount of available capacity during a planning interval. Note that in practice durations or amounts of resources used by services may vary over time and that system noise and distortions exist, which might lead to inequality

between the real utilization and modeled allocation. Parameters, such as resource utilization, can be measured by operating system monitoring tools such as Microsoft's perfmon.

Expected Resource Demand

In addition to the service requests that are already in the system, we also take into account new service requests (e.g., videos and audios) and their resource consumption that we expect in the planning period. We limit the expected duration of future requests to the end of the planning period $t_k + t_i$. (see Figure 3), because resource consumption after $t_k + t_i$ is not relevant to the decision at hand. The factor q_{ei} describes the percentage of resource consumption of service requests within the planning period l_k.

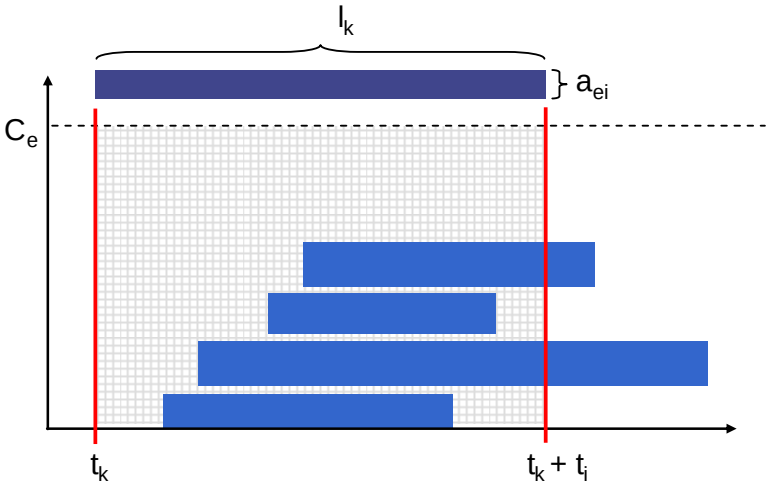

Figure 3 Expected Resource Demand

DLPc describes the problem formulation for an incoming service request k including the estimates for parameters C_e and q_{ei}. D_i describes the expected demand for service i per time unit.

$$
\begin{aligned}
\max \quad & \sum_{i \le I} r_i x_i & & \\
s.t. \quad & \sum_{i \le I} q_{ei} a_{ei} t_{ei} x_i \le l_k C_e & \forall e \le E & \\
& x_i \le l_k D_i & \forall i \le I & \qquad \text{(DLPc)} \\
& x_i \in Z_+ & \forall i \le I &
\end{aligned}
$$

D_i and the resource consumption a_{ei} are stochastic variables. In order to get a good estimate for service demand, we draw on time series forecasting. In our experiments, we have used single or double exponential smoothing. Load tests are necessary to estimate the resource allocation coefficients a_{ei} for resources such as CPU time, bandwidth or memory used [30] [31, 32]. Load testing tools generate artificial workloads on the system. During the tests, server components are monitored and performance metrics (e.g., response time, latency, throughput, late send rate) are measured. Data obtained in this way can be used to identify the resource requirements of single service requests and capacity available.

Experimental Evaluation

In our experimental evaluation we compared DLPc to static admission control rules and used revenue generated under different overload scenarios as a benchmark. We have used two scenarios in order to get first experimental results. In our experiments we assumed a portfolio of four service types: VoD Premium (new movies and blockbusters), VoD Standard (older movies), AoD Premium (top 100 music albums) and AoD Standard (others), using a single media streaming server. The length of a video stream was 90 minutes; the length of an audio stream was 45 minutes. Media files were streamed continuously without stopping and restarting, rewinding, fast-forwarding, jumping etc. Table 1 summarizes service prices, durations and resource requirements concerning the bottleneck resources network and disk. Bottleneck analyses and the determination of resource requirements have been done by load tests as described in the previous section. As load generator, Microsoft's Windows Media Load Simulator for Windows Media Services [33] has been used.

Service Type	Price (€)	Length (min)	Network usage (%)	Hard Disk usage (%)
VoD-Premium	3.–	90.00	$\frac{4262\,Kbit/s}{950000\,Kbit/s}$	$\frac{1}{269}$
VoD-Standard	1.–	90.00	$\frac{4262\,Kbit/s}{950000\,Kbit/s}$	$\frac{1}{269}$

AoD-Premium	0,03	45.00	$\frac{129\ Kbit/s}{950000\ Kbit/s}$	$\frac{1}{6333}$
AoD-Standard	0.01	45.00	$\frac{129\ Kbit/s}{950000\ Kbit/s}$	$\frac{1}{6333}$

Table 1 Media Services Portfolio

We conducted a series of load tests. We generated load only for video files to determine the maximum number of parallel video streams the media server was able to transmit without facing performance problems. To detect performance problems the so called *Late Send Rate* provided by the Windows Media Services was monitored. Windows Media Services computes the amount of data to send per connection and time interval required for continuous media streaming to each connected client. This value is compared to the transmitted data rate. If the transmitted data rate is lower than the data rate required, i.e. data is sent too late.

When streaming videos, the bottleneck resource was the network connection. When streaming audio files, the hard disk throughput was the first bottleneck. The reason for the lower hard disk throughput is that when transmitting a huge amount of low bandwidth streams, the magnetic write/read head of a hard disk is repositioned with high frequency as the audio is streamed from many different files. Note that, depending on the media streaming software used, the files streamed, the hardware used, as well as system configurations, different server resources might become bottlenecks. Load tests were also used to derive the proportion of bandwidth and hard disk utilization that a single service request consumes. For example, for video streams, the resource allocation coefficient of network capacity was 1/222 as the server was able to stream 222 parallel videos (see Table 1).

For the experimental evaluation we set up a test infrastructure consisting of multiple streaming clients, a media server and the admission control server as gateway to the media server. As streaming server software Windows Media Services was used with Windows 2003 64bit Enterprise Edition as the underlying operating system. The media server was installed on a 3.8 GHz Intel Pentium

64bit computer with 2GB DDR2 SDRAM main memory. The server was connected to the switch via a 1 Gbit/s Fast Ethernet connection. As storage a RAID-5-cluster of Fast-ATA hard discs with 7200rpm and 16 MB Cache on each hard disc was part of the environment. Streaming clients have been installed on 2 GHz Intel Pentium 4 machine with 512 GByte main memory. As streaming player software we used Nullsoft's Winamp Player, Version 5.3. The players have been configured to buffer 1 second of a media stream before playing the media. The admission control server was installed on a 2 GHz Pentium 4 PC with 1 GByte main memory. The admission controller was implemented in Java running on Sun's Java Runtime Environment 1.5. The clients were connected to the media server, respectively the admission control server, via 100 Mbit/s Ethernet connections to the switch.

In order to speed-up, automate and control the demand behavior of the streaming clients we initiated streaming requests by the client part of our admission control tool (see Figure 4). The tool generates workload by starting connection requests in predefined intervals, according to predefined demand distribution or playback of log files. Furthermore, the tool allows for defining and running experiments and to analyze experimental results. Streaming requests have been initiated by invoking a method of the Admission Controller Software. If the Controller accepts a streaming request, a media player on a client machine was initiated to request a media file that was then streamed to the client.

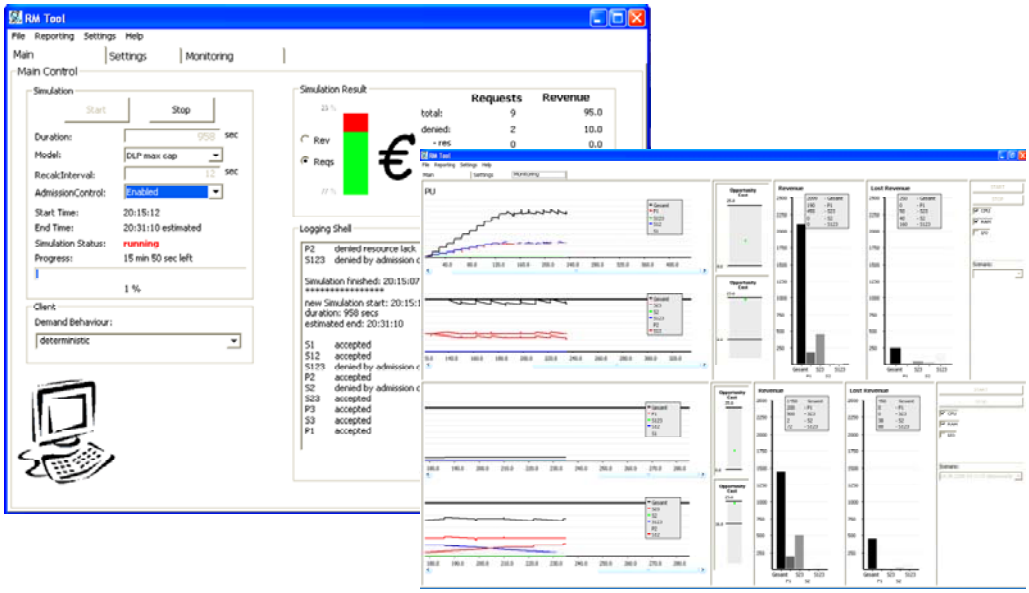

Figure 4 Screenshots of the Simulation Tool

Demand Scenarios

The following admission control policies shown in table 2 have been evaluated experimentally based on the scenario in Table 1 and on different demand scenarios.

DLPc	As described in the previous section
Simple overload control	Connection requests are accepted as long as the maximum bandwidth allocation that guarantees continuous streams is reached. This method is available in off-the-shelf media streaming products.
Threshold based Control	The threshold concerning bandwidth allocation was set to 80%. If the workload demand exceeds this value, only Premium services are accepted. If the maximum bandwidth allocation is reached, all service requests are denied.

Table 2 Admission Control Strategies

We have used two types of demand scenarios. *Flat demand*, where we assume a stable demand, but different combinations of audios and videos. *Variable demand*, where we generated a demand profile with demand peaks. The data generated follows the patterns of media streaming demand that has been described in

previous studies [34]. The duration of each experiment (demand scenario) was set to 720 minutes.

Flat Demand

We define a maximum workload as one that uses 100% of the available capacity. A workload level of 110% means, the capacity of a bottleneck resource has to be increased by at least 10% to serve all incoming requests. In our experiments, we simulated flat demand generating workload levels of 110%, 130% and 150% for 720 minutes.

As these workloads can be generated by different service demands for audio and video files, we repeated each experiment for the three level of workload (110%, 130%, and 150%) with the following Audio Video Request Mixes (AVMs), defining the bandwidth usage ratio of audio and video requests: Only audio, 75 audio and 25 % video, 50 % audio and 50 % video, 25 % audio and 75 % video, and only video. We assumed demand to be 50% of the requests for Premium and 50% for Standard audios and videos.

Variable Demand

In the variable demand scenario, we modeled the demand as was described in previous studies in this field (see Figure 5). We generated mixed demand consisting of audio and video requests. We assumed an AVM of 1:3 and a demand curve for server bandwidth as shown in Figure 5 as an example. The grey line shows the forecasts using simple exponential smoothing with a factor of 0.7. This forecasting method is clearly suboptimal and can easily be improved by double exponential smoothing or more advanced forecasting methods, however, we wanted to analyze the impact of less then optimal forecast. Again, we assumed demand for audio and video by 50% Premium and 50% Standard service requests.

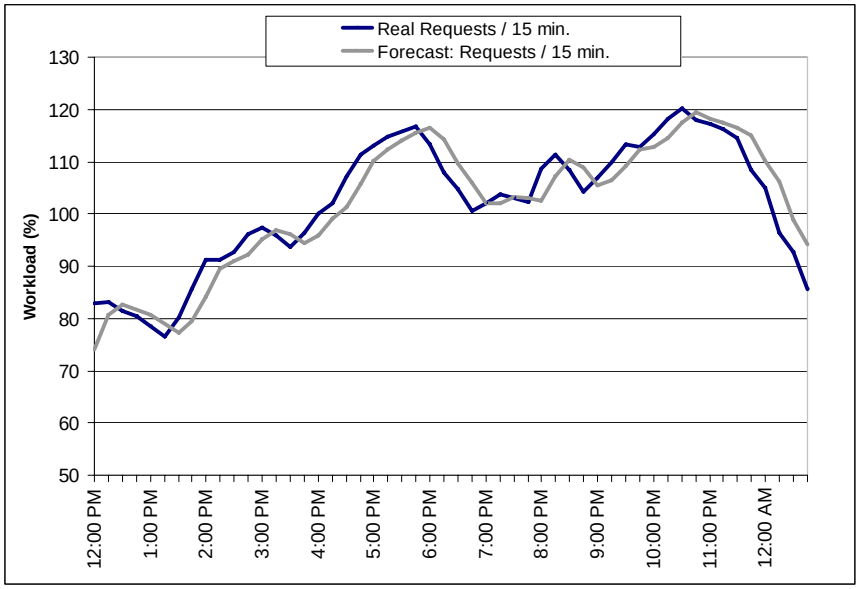

Figure 5 Variable Workload Demand

Experimental Results

Figure 6 – Figure 8 describe the average revenue based on the different admission control policies for selected demand scenarios. Revenue is described as percentage of the revenue "theoretically" possible if all requests could have been accepted. The admission control policies *Threshold Based Control* and *Simple Overload Control* use bandwidth allocation as indicator of high load, as measured by the Windows Media Services software. To avoid overload, the earliest bottleneck determines the maximum server capacity, i.e. the bottleneck disk throughput when streaming only audio. DLPc has been parameterized with the capacity limits of network bandwidth and hard disk throughput measured during the load tests.

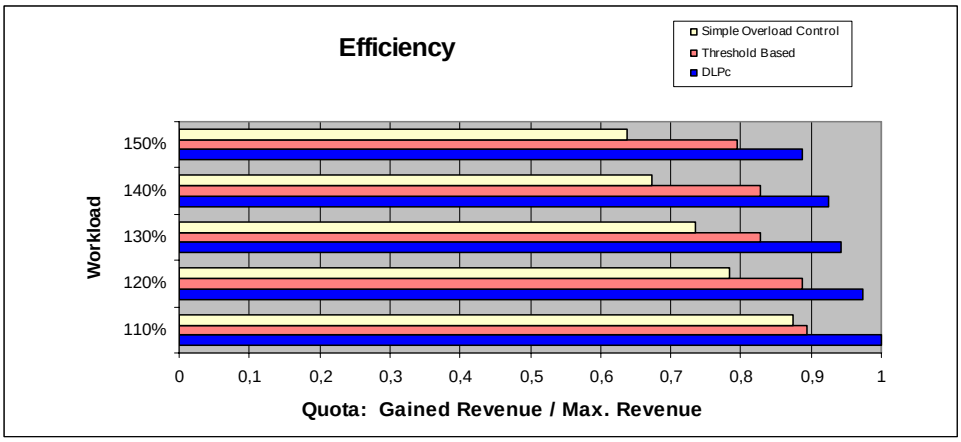

Figure 6 AVM1:3, 130% Workload, Flat Demand

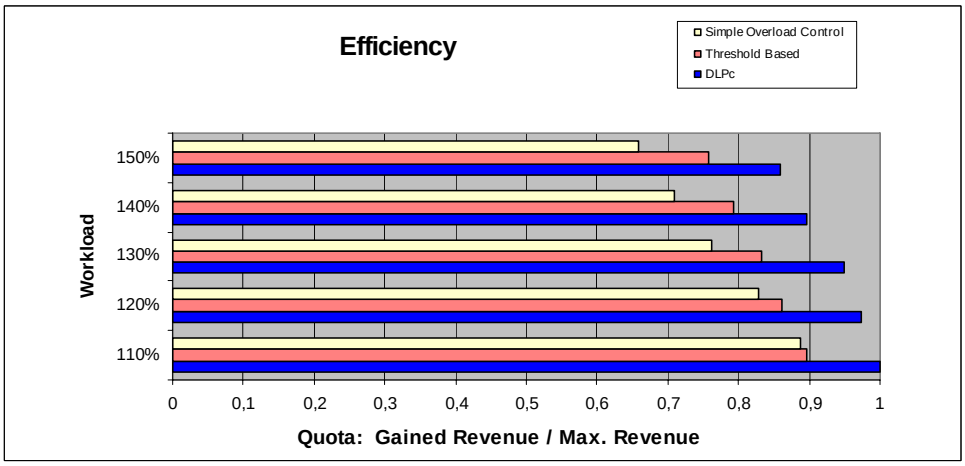

Figure 7 Only Video, 130% Workload, Flat Demand

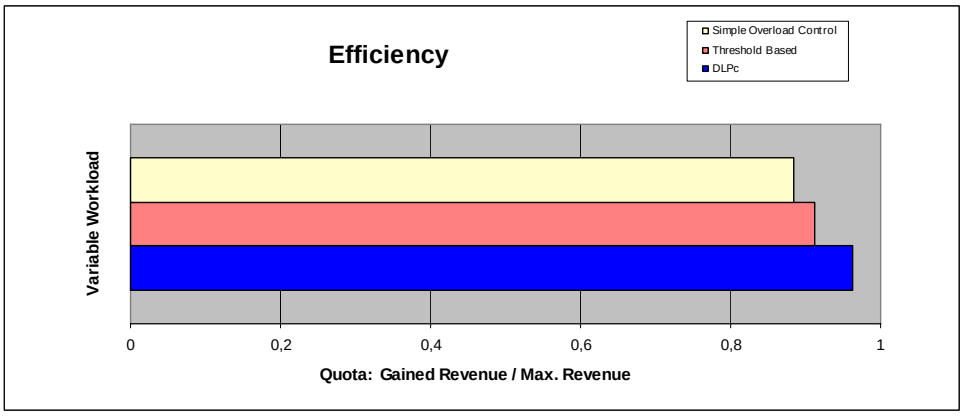

Figure 8 AVM 1:3, Variable Demand

Results show that DLPc dominated the static approaches for each experiment under flat demand and variable demand. For example, with an AVM of 1:3, flat

demand and a workload level of 130%, the revenue based on the data in table 1 was 424 Euro in the time period of 720 minutes when using DLPc, 372 Euro when using the threshold based approach and 330 Euro when using the simple overload control (see Figure 6). The dominance of DLPc is based on the fact that it takes multiple resources into account and reserves capacity for future services requests with higher revenue. Reservation works well if demand is flat, as forecasted demand equals real future demand and the right amount of capacity for expected higher value requests is reserved.

In case of variable demand, forecasting inaccuracies result in biased parameters for the optimization model. In the variable demand scenario, the total number of requests for video and audio streams was 4786. While simple overload control accepted a total sum of 4431 requests, the threshold based approach accepted only 3631 requests, and DLPc 3722 requests. Even when using the rather simple exponential smoothing forecasting, DLPc performed significantly better than static approaches. The results could be repeated with similar artificial time series. A more exhaustive set of experiments with different, possibly real-world demand scenarios and different media portfolios is planned for our future work.

Conclusions and Future Work

Admission control is a central issue for loosely coupled services in the emerging service oriented computing landscape, where service demand is often hard to predict. In this paper, we have focused on the challenges of media streaming services. We have described DLPc, an admission control model and a respective controller for Media on Demand streaming services. While existing admission control approaches are mostly following predefined static rules and treat all incoming request equal, DLPc considers service differentiation and is adaptive as it updates its demand forecast during operation. DLPc addresses the problem of allocating multiple scarce resources. The DLPc controller rejects services early in order to reserve resources for high-revenue services. In our experiments, DLPc achieved significantly higher revenue compared to alternative static methods. In our future work we plan to do more extensive sensitivity analyses with respect to different demand behavior, infrastructural assumptions, or service portfolios. We

also plan to test the DLPc controller in the field and explore admission control problems in other domains.

Acknowledgement

This work was accomplished in collaboration with Siemens Business Services (SBS), one of the largest IT Service Providers in Europe. We thank Siemens Business Services for their technical and financial support.

References

1. Frost&Sullivan (2005) *World Media Streaming Platform Markets*, Frost & Sullivan. Palo Alto, USA
2. Cherkasova, L., W. Tang, and A. Vahdat (2004) *MediaGuard: A Model-Based Framework for Building QoS-aware Streaming Media Services. HP Labs Report No. HPL-2004-25*.
3. Cherkasova, L. and L. Staley (2003) *Measuring the Capacity of a Streaming Media Server in a Utility Data Center Environment* Internet Systems and Storage Laboratory, HP Laboratories. Palo Alto, USA.
4. OGC (2002) *ITIL Best Practice for Service Delivery*. Fourth Edition, Norwich: The Stationary Office.
5. OGC (2003) *IT Infrastructure Library (ITIL)*. [World Wide Web Resource, cited 2005 2005-09-18]; Available from: http://www.itil.co.uk/.
6. Xia, Z., et al. (2006) *An integrated admission control scheme for the delivery of streaming media.* Journal of Parallel and Distributed Computing, 66(3): p. 334-344.
7. Yubing Wang, M.C., and Zheng Zuo (2001) *An empirical study of realvideo performance across the Internet.* In: *ACM SIGCOMM Internet Measurement Workshop*. San Francisco, USA.
8. Kim, R.Y., T. Manas, and K.S.U. Pramod (2005) *Policy-Based Admission Control And Bandwidth Reservation For Future Sessions*, in *European Patent Office*.
9. Vin, H., A. Goyal, and P. Goyal (1994) *A statistical admission control algorithm for multimedia servers*. In: *International Multimedia Conference*. San Francisco, USA.
10. Chen, I.-R. and C.-M. Chen (1996) *Threshold-Based Admission Control Policies for Multimedia Servers.* The Computer Journal, 39(9): p. 757-766.
11. Cheng, S., C. Chen, and I. Chen (2003) *Performance evaluation of an admission control algorithm: Dynamic threshold with negotiation.* In: Perform. Eval,. 52(1): p. 1-13.
12. Acharya, S., et al. (2000) *Characterizing User Access to Videos on the World Wide Web*. In: *ACM / SPIE Multimedia Computing and Networking*.
13. Almeida, J.M., et al. (2001) *Analysis of Educational Media Server Workloads*. In: *11th International Workshop on Network and Operating System Suport for Digital Audio and Video*.
14. Cherkasova, L. and M. Gupta (2002) *Characterizing Locality, Evolution and Life Span of Accesses in Enterprise Media Server Workloads.* In: *12th International Workshop on Network and Operating System Support for Digital Audio and Video ACM NOSSDAV*.
15. Chen, X., P. Mohapatra, and H. Chen (2001) *An Admission Control Scheme for Predictable Server Response Time for Web Accesses*. In: *World Wide Web Conference*.
16. Wang, Y., M. Claypool, and Z. Zuo (2001) *An empirical study of realvideo performance across the Internet*. in *ACM SIGCOMM Internet Measurement Workshop*. San Francisco, USA.
17. Ge, Z., J. Ping, and P. Shenoy (2002) *A Demand Adaptive and Locality Aware (DALA) Streaming Media Server Cluster Architecture.* In: *International Workshop on Network and Operating System Support for Digital Audio and Video*. Miami, USA.

18. Cherkasova, L. and L. Staley (2003) *Building a Performance Model of Streaming Media Applications in Utility Data Center Environments*. In: *International Symposium on Cluster Computing and the Grid (IEEE Computer Society)*.
19. Apple Computer (2003) *QuickTime Streaming Server 5.0 Administration*, Apple Computer, Inc.
20. RealNetworks (2005) *Helix Server Administration Guide* [cited 2006 September, 26, available from: http://service.real.com/help/library/guides/HelixServerWireline/wwhelp/wwhimpl/js/html/wwhelp.htm.
21. Microsoft (2006) *Windows Media Services 9 Series* [cited 2006 September, 12, available from: http://www.microsoft.com/windows/windowsmedia/forpros/server/server.aspx.
22. Adobe (2006), *Flash Media Server 2 Documentation* [cited 2006 September, 15, available from: http://download.macromedia.com/pub/documentation/en/flashmediaserver/2/fms_pdfs.zip.
23. Kwon, J. and H. Yeom (2000) *An Admission Control Scheme for Continuous Media Servers Using Caching*. In: *Int'l Performance, Computing and Communication Conference (IPCCC)*. Phoenix, USA.
24. Vin, H.M., A. Goyal, and P. Goyal (1994) *An observation-based admission control algorithm for multimediaservers*. in *Multimedia Computing and Systems*. Boston, USA.
25. Welsh, M., D. Culler, and E. Brewer (2001) *SEDA: An architecture for well-conditioned, scalable Internet services*. in *18th Symposium on Operating Systems Principles*. Chateau Lake Louise, Canada.
26. Welsh, M. and D. Culler (2003) *Adaptive Overload Control for Busy Internet Servers*. In: *4th Usenix Conference on Internet Technologies and Systems (USITS)*.
27. Welsh, M. and D. Culler (2002) *Overload management as a fundamental service design primitive*. In: *Tenth ACM SIGOPS European Workshop*. Saint-Emilion, France.
28. Williamson (1992) *Airline Network Seat Inventory Control - Methodologies and Revenue Impacts*, in *Department of Aeronautics and Astronautics*. MIT, USA.
29. Talluri, K.T. and G.J. van Ryzin (1999) *An Analysis of Bid-Price Controls for Network Revenue Management.* Management Science, 44 (1577-1593).
30. Brandl, R. (2006) *Reinraum-Messungen zur Verrechnung von IT-Anwendungen*. in *Multikonferenz Wirtschaftsinformatik*. Passau, Germany.
31. Nagaprabhanjan, B. and V. Apte. (2005) *A tool for automated resource consumption profiling of distributed transactions*. in *Second international conference, ICDCIT*. Bhubaneswar, India.
32. Kounev, S. and A. Buchmann (2003) *Performance Modelling and Evaluation of Large-Scale J2EE Applications*. In: *29th International Conference of the Computer Measurement Group (CMG)*.
33. Microsoft (2006) *Windows Media Load Simulator for Windows Media Services 9 Series*. [cited 2006 Oktober, 9, available from: http://www.microsoft.com/windows/windowsmedia/forpros/serve/tools.aspx.
34. Yu, H., et al. (2006) *Understanding User Behavior in Large-Scale Video-on-Demand Systems*. In: *EuroSys*.