QoS Control Challenges for Multimedia Consumer Terminals

Clemens C. Wüst[1], Reinder J. Bril[2], Christian Hentschel[3], Liesbeth Steffens[1], Wim F.J. Verhaegh[1]

1. Philips Research Laboratories, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands
2. Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
3. Brandenburg University of Technology Cottbus, P.O. Box 10 13 44, 03013 Cottbus, Germany

clemens.wust@philips.com, r.j.bril@tue.nl, christian.hentschel@tu-cottbus.de,
liesbeth.steffens@philips.com, wim.verhaegh@philips.com

Abstract

In this paper, we present QoS control challenges for multimedia consumer terminals based on an application execution model and a QoS resource management framework. In the context of this framework, we briefly recapitulate earlier work aimed at QoS control for high-quality video processing. By relaxing a number of assumptions in that work, and by considering the work in relation to the framework and the applied video algorithms, new control challenges are identified.

Keywords: Soft real-time, overload, multimedia, Quality of Service (QoS).

1 Introduction

Consumer terminals are gradually evolving from straightforward terminals of a video broadcast network (TV sets) and a communication network (telephones) to interactive multimedia consumer terminals (MCTs), and beyond that to elements in an in-home network, or even an ambient intelligent environment [1]. Moreover, there is a trend towards systems where significant parts of the media processing are carried out in software.

Subjects addressed are QoS in software-based MCTs, such as digital TV sets, digitally improved analog TV sets and set-top boxes. The basic media in MCTs are high-quality audio and video. If the basic media processing functions are scalable, they can create room for other media processing functions at little or no extra cost [2]. The challenge of QoS for MCTs is in finding a QoS approach that can primarily be applied to high-quality video, and also supports other media, such as 3D graphics. The main focus of this paper will be on QoS for high-quality video.

Media processing in software is required to be cost-effective, while preserving qualities as robustness, predictability, and stability. Cost-effectiveness requires that the resources are allocated, provided, and used effectively, towards the goal of maximizing the overall QoS. Within Philips Research, this challenge has been addressed by a multi-disciplinary team. This resulted in a QoS approach that combines *application adaptation* and *QoS-based resource management* [2].

Scalable applications can operate at different quality levels, allowing run-time trade-offs between output quality and resource usage by controlling the *operational quality*. Examples of scalable applications from the video domain are given in [3] and from the 3D graphics domain in [7].

The basis of the approach is constituted by a software framework for QoS-based resource management, that has been designed as a combination of a *multi-layer control hierarchy* and a *reservation-based resource manager* [10].

The work presented in [13] is aimed at QoS control for high-quality video processing, assuming a basic, simplified, real-time processing model, and a rudimentary framework. This work resulted in an approach that allows close to average-case resource allocation to a single video processing task, based on asynchronous, scalable processing, and QoS adaptation. In this paper, we focus on QoS control challenges for MCTs, starting from a brief description of our application execution model and our QoS resource management framework.

The paper is organized as follows. In Sections 2 and 3, we first discuss our application execution model and QoS resource management framework. Next, in Section 4 we briefly recapitulate our basic approach to QoS control for high-quality video processing. In Section 5 we identify a number of QoS control challenges for MCTs. Finally, in Section 6 we state the conclusions.

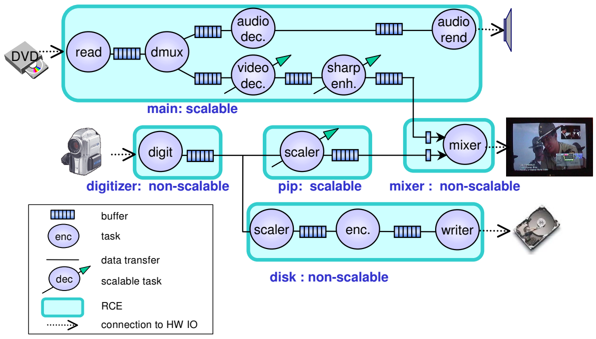

Figure 1. Application execution model.

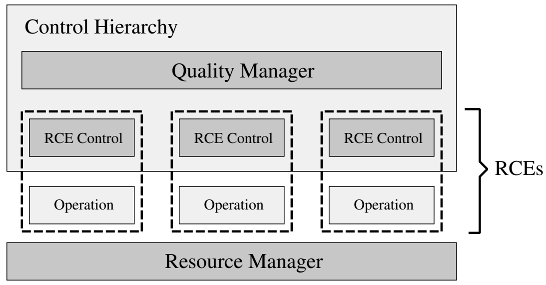

Figure 2. Simplified view of the QoS resource management framework.

2 Application Execution Model

Figure 1 depicts the application execution model [10] of a demonstrator that has been built to show the feasibility of our QoS approach and to test the concept. The demonstrator was earlier presented in [4]. The figure contains five so-called *resource consuming entities* (RCEs), constituting a set of media applications. An application is given by the combination of one or more RCEs, and an RCE may be used for multiple applications. Note that an RCE may contain multiple tasks, and that a task may, but need not, be scalable. An RCE is scalable if it contains one or more scalable tasks. The quality levels of the different scalable tasks are combined into a meaningful set of quality levels for the RCE. A scalable RCE provides an interface to higher layers in the control hierarchy for selecting a coarse-grained quality level. A coarse-grained quality level consists of a number of fine-grained quality levels, which roughly provide the same coarse-grain output quality. An RCE controller within the RCE makes sure that the RCE produces an acceptable result, given the resource budget provided by the resource manager, by controlling the fine-grained quality level.

3 QoS Resource Management Framework

As described before, the basis of our QoS approach is constituted by a software framework for QoS-based resource management. Figure 2 gives a simplified view of this framework. Note that the RCEs reside at the bottom of the control hierarchy. The RCEs are responsible for both effective and efficient resource usage for media processing.

The control hierarchy is responsible for effective, dynamically adjusted, resource allocation. Overall effectiveness is to a large extent realized by dynamically maximizing the overall QoS, using a global notion of *utility*. The optimization, performed by the quality manager (QM), is based on the momentarily available mapping from coarse-grained quality levels to resource needs of RCEs, and also takes the relative importance of RCEs into account, using a model similar to the one described in [8]. By selecting a coarse-grained quality level for an RCE, the QM determines the operational setting at which that RCE is expected to run. In [10], we distinguished four layers of QoS control in the hierarchy, three layers of global QoS control, one of which is the QM, and one layer of local (RCE specific) QoS control. The control hierarchy addresses stability by choosing appropriate time scales for the different control layers, and appropriate, non-interfering, control algorithms.

The resource manager (RM) is responsible for effective and efficient resource provision. The RM addresses robustness and predictability by providing *guaranteed resource budgets*, which are based on *resource reservation*. Resource reservation [9, 12] is recognized as a basis for QoS resource management [8].

4 Basic RCE Control

In [13], we considered a single-threaded, soft real-time, scalable video processing task, with highly fluctuating, data dependent, resource requirements. The task provides a limited number of quality levels, by means of which picture quality can be traded for resource usage at the level of individual frames [3]. The processing time for a frame depends on both the selected quality level and the data complexity of the frame. Frames to be processed are retrieved from an input queue, and processed frames are placed into an output queue. An input process periodically inserts new frames into the input queue, and an output process consumes processed frames from the output queue, with the same period. Hence, we assume that the input and output frame rates are the same. The input process and output process are synchronized with a fixed latency. We assume a latency larger than the period, which allows the task to work ahead by means of asynchronous processing, at cost of adding input and output buffers. The task has to operate within a given fixed periodic budget, which is less than the worst-case requirement of the task. To handle deadline misses, a work-preserving approach is applied.

Before processing a frame, a controller selects the quality level at which the frame is processed. We trade-off three aspects of user-perceived quality in a single QoS measure: deadline misses, the quality level at which frames are processed, and changes in the quality level between successive frames. This approach leads to the QoS control problem of selecting a quality level for each frame to be processed, to maximize the QoS measure.

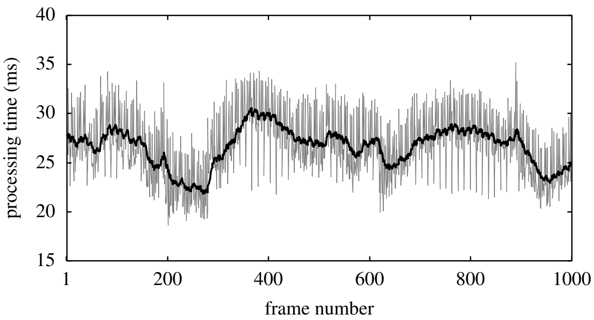

Figure 3. The load for decoding a sequence of MPEG-2 frames, and the corresponding structural load as viewed by the controller (black line).

The QoS control strategies presented in [13] are able to accommodate the normal stochastic (or short-term) load fluctuations as well as data-dependent structural (or long-term) load fluctuations. Figure 3 shows the load for decoding a sequence of MPEG-2 frames, and the corresponding structural load as viewed by the controller. To handle stochastic load fluctuations, the control problem is modeled as a Markov Decision Process, which is a general approach for solving discrete stochastic decision problems. A static solution based on off-line optimization is presented, as well as a dynamic solution based on Reinforcement Learning. To handle structural load fluctuations, a scaled budget enhancement is used.

5 QoS Control Challenges

In this section, we identify QoS control challenges for MCTs by relaxing assumptions in the basic model for RCE control, and by considering its relation to the QoS resource management framework and the controlled video algorithms.

5.1 RCE Control

In the basic model for RCE control, we assumed a fixed processing-time budget for the scalable video processing task, and we implicitly assumed that the task's input data has a fixed quality. In many practical situations, however, these two assumptions are not valid.

As shown in Figure 1, an RCE may consist of multiple tasks. A budget assigned to an RCE is shared by all tasks in the RCE. Whenever other tasks in the same RCE have a data-dependent load, we may no longer assume a fixed budget for the scalable task. Apart from that, a budget is usually provided as a *minimum guarantee* on the processing time available for the task in a period. Moreover, this minimum guarantee is usually not strict, but approximate, subject to some stochastic distribution. Instead of knowing the available budget till the next deadline, the controller will have to make a best guess, which again requires some kind of a decision algorithm.

Depending on the application execution model, the quality of the input data is not always fixed. This is especially true for distributed systems and for heterogeneous multiprocessor systems on a chip. In both cases, the input data for the scalable task is the output data of a task running on a different computational unit, and possibly sent over an unreliable connection. In [6], an initial approach to handling a fluctuating input quality in the context of layered MPEG-2 for wireless networks is presented. It is assumed that not all layers are received, due to unreliable transfer of data. A more general approach to handling fluctuating input quality will be needed.

5.2 QoS Resource Management Framework

To achieve stability in the control hierarchy, it is important to separate the normal stochastic load fluctuations of RCEs from structural ones. In [13], a simple feedback filter is used by the controller to determine the structural load. This structural load fluctuates too much to be used by the QM, which makes global QoS control decisions at a much lower frequency. In the view of the QM, the structural load remains constant for a much longer period of time. Deciding that a load change is structural from a QM point of view is a non-trivial problem.

Each RCE has its own local QoS measure. The QM requires a single global QoS measure, which has to be optimized for the system as a whole. Hence, the different local QoS measures have to be mapped to that single global QoS measure. Creating this mapping is a challenge in its own right, in addition to the challenge of global QoS optimization.

Effectively re-allocating unused reservations is an essential part of any approach that aims at optimizing systems with scarce resources. Currently, the RM does not explicitly control the re-allocation of unused time over the different RCEs. A value-based approach [11] might be used to determine which RCE can make the best use of the additional time. Typically, this would be the RCE that currently has the most difficulty in realizing its intended QoS, as determined by the QM. The re-allocation control strategy would work at the same time scale as the RCE controller, and it should require very little run-time overhead. To avoid instability, the combination must be very carefully managed by the QM.

5.3 High-Quality Video

In our model for RCE control, the QoS measure is based on rewards for the selected quality levels, and penalties for both deadline misses and quality-level changes. The rewards and penalties used in [13] were based on input from video-domain experts. Before the control algorithm can be deployed in a real system, these rewards and penalties would have to be determined by extensive user-perception experiments. Moreover, the resulting QoS control decisions have to be validated in user-perception experiments as well.

So far, we used MPEG decoding as a vehicle for research, because it was the main scalable algorithm available at the time. To keep our approach applicable to other video algorithms as well, we relaxed some MPEG idiosyncrasies, such as frame types (I, P, B) and the difference between the decoding order and the presentation order of frames [5]. Tuning RCE control strategies to specific scalable video algorithms is a topic to be addressed.

6 Conclusion

In [13], we presented QoS control strategies for a soft real-time video processing task, based on probabilistic techniques. That work assumed a basic QoS resource management framework and a basic real-time processing model. In this paper, we gave a description of the framework and of a corresponding application execution model. Given that description, we identified a number of QoS control challenges for multimedia consumer terminals.

For the extensions of our control approach, we aim at enhancing our existing QoS control strategies. As a working hypothesis, we assume that probabilistic techniques also apply to other layers in the control hierarchy. The particular order of the steps to be taken strongly depends on the composition of the multi-disciplinary team that will address these control challenges.

References

[1] E. Aarts, R. Harwig, and M. Schuurmans. Ambient intelligence. In P.J. Denning, editor, *The invisible future: The seamless integration of technology into everyday life*, pp. 235–250, McGraw-Hill, Inc., New York, NY, USA, 2001.

[2] R.J. Bril, C. Hentschel, E.F.M. Steffens, M. Gabrani, G.C. van Loo, and J.H.A. Gelissen. Multimedia QoS in consumer terminals. In *Proc. IEEE Workshop on Signal Processing Systems (SIPS)*, pp. 332–343, 2001.

[3] C. Hentschel, R. Braspenning, and M. Gabrani. Scalable algorithms for media processing. In *Proc. International Conference on Image Processing (ICIP)*, pp. 342–345, 2001.

[4] C. Hentschel, R.J. Bril, Y. Chen, R. Braspenning, and T.-H. Lan. Video quality-of-service for consumer terminals – a novel system for programmable components. In *Digest of technical papers International Conference on Consumer Electronics (ICCE)*, pp. 28–29, 2002.

[5] D. Isović and G. Fohler. Quality aware MPEG-2 stream adaption in resource constrained systems. In *Proc. 16th Euromicro Conference on Real-Time Systems (ECRTS)*, pp. 23 – 32, 2004.

[6] D. Jarnikov, P.D.V. van der Stok, and C.C. Wüst. Predictive control of video quality under fluctuating bandwidth conditions. In *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, 2004.

[7] G. Lafruit, L. Nachtergale, K. Denolf, and J. Bormans. 3D computational graceful degradation. In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, Vol. 3, pp. 547–550, 2000.

[8] C. Lee, J. Lehoczky, R. Rajkumar, and D. Siewiorek. On Quality of Service optimization with discrete QoS options. In *Proc. 5th IEEE Real-Time Technology and Applications Symposium (RTAS)*, pp. 276–286, 1999.

[9] C.W. Mercer, S. Savage, and H. Tokuda. Processor capacity reserve: operating system support for multimedia applications. In *Proc. International Conference on Multimedia Computing and Systems (ICMCS)*, pp. 90–99, 1994.

[10] C.M. Otero Pérez, L. Steffens, P. van der Stok, S. van Loo, A. Alonso, J.F. Ruíz, R.J. Bril, and M. García Valls. QoS-based resource management for ambient intelligence. In T. Basten, M. Geilen, and H. de Groot, editors, *Ambient Intelligence: Impact on Embedded System Design*, pp. 159–182. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.

[11] D. Prasad, A. Burns, and M. Atkins. The valid use of utility in adaptive real-time systems. In *Real-Time Systems*, Vol. 25, pp. 277–296, 2003.

[12] R. Rajkumar, K. Juvva, A. Molano, and S. Oikawa. Resource kernels: A resource-centric approach to real-time and multimedia systems. In *Proc. SPIE, Vol. 3310, Conference on Multimedia Computing and Networking*, pp. 150–164, 1998.

[13] C.C. Wüst, L. Steffens, R.J. Bril, and W.F.J. Verhaegh. QoS Control Strategies for High-Quality Video Processing. In *Proc. 16th Euromicro Conference on Real-Time Systems (ECRTS)*, pp. 3–12, 2004.