QoS concept for scalable MPEG-4 video object decoding on multimedia (NoC) chips

Citation for published version (APA):
Pastrnak, M., With, de, P. H. N., & Meerbergen, van, J. (2006). QoS concept for scalable MPEG-4 video object decoding on multimedia (NoC) chips. *IEEE Transactions on Consumer Electronics*, *52*(4), 1418-1426. https://doi.org/10.1109/TCE.2006.273165

DOI:
10.1109/TCE.2006.273165

Document status and date:
Published: 01/01/2006

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

QoS Concept for Scalable MPEG-4 Video Object Decoding on Multimedia (NoC) Chips

Milan Pastrnak, Peter H. N. de With, *Senior Member*, IEEE, Jef van Meerbergen, *Senior Member*, IEEE

Abstract — *Scalable implementations of multimedia applications offer increased flexibility in mapping those applications onto the executing platform used in a consumer product. In this paper, we describe a hierarchical Quality-of-Service (QoS) model for managing multimedia applications running on a multiprocessor Systems-on-Chip (SoC). First, we present the possible scalability of an MPEG-4 arbitrary-shaped video decoder with respect to computational and communicational resources. Second, we provide a novel model for QoS management based on the principles of predictable mapping and run-time information on the resource utilization. We demonstrate the QoS framework by mapping of an MPEG-4 arbitrary-shaped decoder on a NoC, employing eight ARM cores with specific monitoring features in the network (e.g. Æthereal NoC). The scalable implementation results in lowering the computational requirements by 26% and communication by 43%. Experiments revealed that the combination results in more than 85% decoded frames of higher quality than in a QoS approach based on the predictable mapping[1] only.*

Index Terms — computation, hierarchical QoS, multimedia NoC, arbitrary-shaped coding.

I. INTRODUCTION

A typical composition of state-of-the-art multimedia applications in present consumer products is based on the simultaneous execution of several stand-alone subsystems of which the results are jointly presented to the user. Current design approaches in modern consumer electronics products are based on two major strategies. On one hand, the focus is on optimizing a system with a single functional requirement, such as a stand-alone DVD player. On the other hand, designers aim at a more general solution offering the combined functionality of several stand-alone applications and further extensibility of the system. An example of the latter case is a smart phone. Our aim is to focus on recent multimedia applications which inherently ask for more general solutions while still providing sufficient control of quality and resource usage.

For cost-efficient consumer (embedded) system design, the platform cost and its resources are bounded, so that quality control among applications and inside applications is inevitable. Quality-of-Service (QoS) management for Systems-on-Chip (SoCs) has been extensively studied, e.g. for MPEG-4 3D graphics, wavelet coding, and related applications [1]. The proposed QoS management approaches compute the resource utilization as an algebraic function of the quality settings, for example by the number of graphical triangles to be processed. Our objective is to provide a QoS architecture that can predict the quality level setting of an application and still re-use non-utilized resources for a temporary QoS increase.

A. Scalable Arbitrary-Shaped MPEG-4 Video Coding

Current video applications are generally processing a single video stream and a single audio stream. Emerging new multimedia applications require more advanced interactivity with the video content and introduce synthetic video objects which enrich the natural video signal. The first standard with the focus on object-based video coding is the MPEG-4 standard [2], in particular the core profile.

In this paper, we focus on a scalable implementation of arbitrary-shaped (AS) MPEG-4 video object decoding, based on the full object-based coding specification. This application is interesting, as it has very dynamic execution time characteristics per frame because of large variations in object size and behavior. Furthermore, when several video objects occur in the picture simultaneously, an equal number of decoding instantiations can be executed in parallel. Each instantiation involves a set of stream-oriented decoding tasks.

We have presented a preliminary model of a streaming-oriented implementation of the AS MPEG-4 decoder in [3]. Depending on the target application of the decoder, a certain quality loss can be acceptable. For example, for a video object containing a soccer ball, the shape and size is more important than a high quality texture. In this paper we present a form of task-level scalability of the texture decoding, providing scalability in different types of resources.

- *Computational scalability* – the decoding chain is modified at run-time for activating/deactivating texture-decoding tasks.
- *Communicational scalability* – the task-level scalability can modify the bandwidth requirements based on the task-to-processor assignment.

[1] This work was supported by European Union via the Marie Curie Fellowship program under the project number HPMI-CT-2001-00150 and by the PROGRESS program of the Dutch Technology Foundation STW through the PreMaDoNa project EES.6390.

Milan Pastrnak was during this research with LogicaCMG Netherlands BV and conducted this work at the Eindhoven University of Technology, in the Video Coding and Architecture group of the Signal Processing Systems Dept. He is now with TOPIC Embedded Systems, P.O. Box 440, 5680 AK Best, The Netherlands (e-mail: M.Pastrnak@tue.nl).

Peter H. N. de With is with Eindhoven University of Technology, VCA group of the Signal Processing Systems Dept. He is also with LogicaCMG Nederland B.V., 5605JB Eindhoven, The Netherlands (e-mail: P.H.N.de.With@tue.nl).

Jef van Meerbergen is with Philips Research Europe-Eindhoven, the Netherlands. He is also with University of Technology, Electronic Systems Group (e-mail: jef.van.meerbergen@philips.com).

Contributed Paper
Manuscript received October 13, 2006

A scalable implementation provides a broader way for the application execution on a resource-limited platform. Furthermore, the scalability improves the finer granularity for resource utilization.

B. *Quality-of-Service (QoS) on Systems-on-Chip*

Several QoS approaches have been reported in literature. For example, economic reservation-based QoS solutions are presented by Bril *et al.* [4]. We have presented our hierarchical QoS proposal in [5]. However, more recent experimental results have shown that pure reservation-based QoS control of the system yields an average efficiency of about 70%. For this reason, we focus on further maximizing the possible output quality by using a reservation-based technique in combination with a best-effort run-time adaptation of the computation. We will show that such a combination indeed improves the picture quality. We present a QoS management model using a NoC run-time monitoring system [6] for run-time adaptation of the computation graph. In the majority of the processed pictures, a switch to a higher quality level of video processing is obtained.

The paper is organized as follows. Section II gives a brief introduction to a predictive computation on the multiprocessor Network-on-Chip (NoC). Section III addresses the problem definition of the recent QoS management. Section IV presents a new combined QoS technique to provide more efficient resource usage. The different levels in scalability of the arbitrary-shaped video decoder are presented in Section V. The experimental framework of combined QoS management on a homogeneous NoC with specific monitoring features in the network (e.g. Æthereal NoC) is discussed in Section VI. Section VII describes the system behavior and experimental results and Section VIII concludes this paper.

II. Predictable Computation on Multiprocessor NoC

The objective is study a multiprocessor system-on-chip with a system architecture that enables a predictable computation. Our leading application is a complex state-of-the-art multimedia algorithm (MPEG-4 shape-texture decoding for individual video objects), which has very dynamic execution time-characteristics per frame, and of which several instantiations can be executed in parallel. Each instantiation is internally composed of several pipelined tasks. It is possible that a set of objects has to be decoded in parallel where each object has its own characteristics and behavior. In our previous work, we motivated the use of a Multi-Processor System-on-Chip (MPSoC) as a target platform for such advanced applications [3]. The efficient mapping of multiple object decoders onto such a platform poses a management problem, requiring QoS control of the platform resources.

The term "network" in Multi-Processor Network-on-Chip (MP-NoCs) refers to the enclosed switch network that is used for global on-chip communication. In our case, we employ a tile-based architecture with distributed memory, as depicted in Figure 1. The MP-NoC platform contains processing tiles, storage tiles, all organized in a networked fashion (e.g. switch network). A processing tile represents a small embedded computer, consisting of one embedded CPU core (e.g. RISC), local memory and specific accelerators. The NoC transports data packets from one tile to another.

In general, NoC can be modeled using SDF graphs, provided that the following constraints on the architecture are satisfied. First, tasks running in parallel on different processors use only the local memories of their processing tiles. Second, the memories are organized in single layers (no caching), or the caches are locked. This provides the predictability of the task computation times. The NoC should provide point-to-point connections with tightly-bounded packet propagation delays. Similar to data edges in SDF graphs, the connections should be independent from each other and they should carry multiple tokens in FIFO order. Such connections can be implemented in NoCs at a reasonable cost [7].

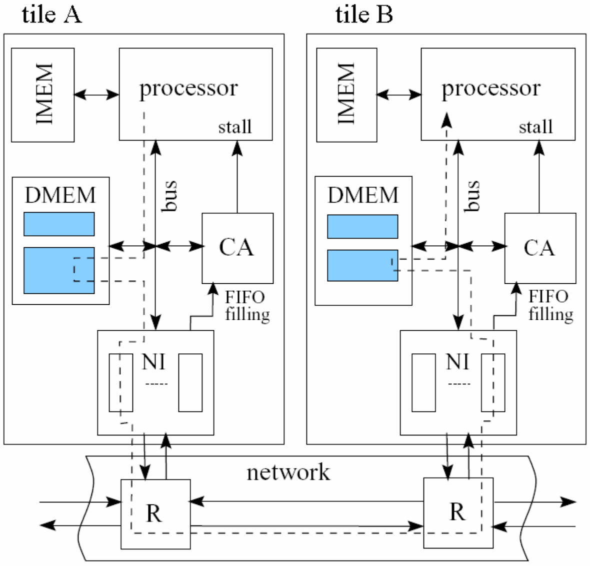

Fig. 1. Tile-based MP-NoC architecture. A processing tile contains an embedded process, local instruction and data memory and a so-called Communication Assist (for more details see [8]).

The mapping of an application in a predictable matter requires a modeling of the application behavior at fine granularity level. To express the multiprocessor-level parallelism in our model, we employ Synchronous Data Flow (SDF) graphs (see e.g. [8]). More precisely, we use a restricted version of the SDF model, called Homogeneous SDF (HSDF).

The computations in an HSDF are represented by the nodes of the HSDF graph, called *actors*. The edges of the graph represent dependencies between actors and carry tokens that are produced and consumed by the actors. Each edge points to the direction of its token flow and may contain a few initial tokens. For preserving consistency, we maintain to call the smallest computation block a *task* instead of an actor.

It is common to distinguish data edges and sequence edges within a graph. Passing of a token through a data edge represents the transfer of a block of data from one task to another. On a sequence edge, the tokens represent events that do not carry data, e.g. the release of space in memory.

Each task waits until there is at least one token at each incoming edge. Then the task performs computations on the contents of the first data token that is available at each data input. The computation takes a well-defined time interval, which only depends on the contents of the input data, called the computation time of the task. When the computations have finished, one token is consumed from each incoming edge and one token is produced to each outgoing edge.

III. Hierarchical Quality-of-Service Management

The separation of responsibilities of the system management is essential to decrease the complexity of the resource assignment. We consider three major classes of the management problem: resource management, inter-application management and intra-application management.

Ref. [9] presents the EUROPA architecture with separated concerns aiming at quality enhancement. There are several examples of QoS system management, like the Padma architecture from [10], $2K^Q$ architecture in [11] and Agile QoS as described in [12]. However, these architectures are limited in some or other directions which cannot fulfill the system requirements of our problem domain.

Our mapping strategy exploits the predictability property of our architecture to enable a deterministic QoS for each job, independent of other jobs. To achieve this, we reserve resources for each particular job in the form of *virtual* processors and *virtual* connections. These virtual processors and connections are run-time assigned to the existing resources of the platform. This abstraction is important to obtain the worst-case model of the resource distribution in the case that each task is mapped to a different processor of the platform, and to keep independency between jobs.

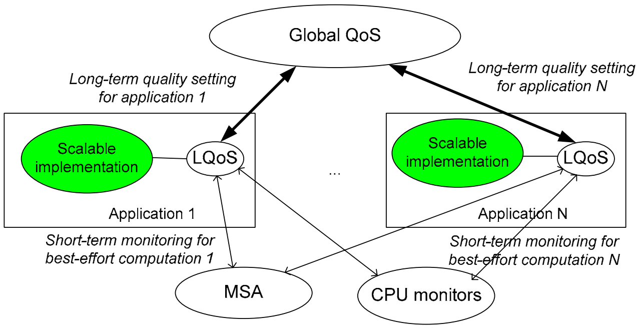
Global QoS
Long-term quality setting for application 1
Long-term quality setting for application N
Scalable implementation
LQoS
Application 1
...
Scalable implementation
LQoS
Application N
Short-term monitoring for best-effort computation 1
Short-term monitoring for best-effort computation N
MSA
CPU monitors

Fig. 2. Intra and inter-application QoS management. A global QoS manager assigns resources at a guaranteed quality level for an application. A local QoS manager adapts the application quality level based on run-time information.

Advanced QoS control requires an accurate estimation of the resource usage. Therefore, we distinguish an *off-line phase* where jobs are mapped to virtual processors to obtain specific application operating points, and a *run-time refinement* of the resource usage based on the current resource-usage status of the system (see Fig. 2). After the assigning quality and invoking the execution, the *adaptation of computation* towards a best-effort result can be enabled. These phases are described in more detail below.

A. *Off-line: job-mapping definition*

The purpose of the intra-job mapping is to generate a set of operating points, which allows to online trade-off between the quality resulting from the job and the resource usage by selecting an appropriate operating point. For each operating point, a certain quality setting is initially assigned. Afterwards, a set of virtual processors and connections are allocated. Different tasks are inserted in sequential order into allocated processes, and the processes are partitioned over the virtual processors. The data transfers between the virtual processors are assigned to the virtual connections. The result of allocation and assignment is a virtual platform for the job, and a network of concurrent communicating processes for the job that is mapped onto the virtual platform. This network is called a *configuration network*.

A major objective of intra-job mapping is to create a virtual platform using minimum resources. On the other hand, the platform should offer enough resources such that the *deadline miss rate* of the job is low enough. Each operating point is defined by a quality setting and a virtual platform with the corresponding configuration network. The quality setting gives only an estimate of the average optimal quality setting for the given mapping. Due to variation of the execution time, at run-time the quality setting is adjusted continuously. We have presented in [13] a cost function similar to the prioritized OS prioritized models with the dynamic modification of the priority, based on the application content.

B. *Run-time: quality negotiation and resource allocation*

The resource manager controls the available physical resources in conjunction with the Global Quality manager, thereby using the operating points which are generated off-line. This works as follows. For a starting job, the Global Quality manager invokes the Local Quality manager (LQoS manager) which chooses an initial quality setting by selecting an operating point. In advance mode, the LQoS manager can activate an estimator module that processes input data parameters for more precise resource requirements per quality level *Q*. Based on off-line measurements of the anticipated quality level *Q*, the manager strives for a quality setting that will satisfy the user.

At this point, the resource manager is of key importance, as it keeps track of the free capacity of all physical resources in the platform. Given a virtual platform, for each virtual processor the manager should find a physical processor with sufficient free capacity. For each virtual connection, free network resources should be found. It may happen that the resource manager cannot accommodate the resources for the new job. If the new job has a high importance, the Global Quality manager may decide to decrease the quality settings of some other jobs to release sufficient resources for the new job.

IV. Problem Statement - Fixed Reservation of Resources

Predictable mapping is an important new paradigm for designing future systems having a broad functionality and the corresponding high amount of parallel execution within such a system. The described approach for a system based only on predictable computing has several drawbacks as listed below.

The decoding of individual video objects is based on several data dependencies. The most well-known dependency is the motion compensation within the MPEG hybrid coding architecture. Both the internal dependency on the intra-coded video frame and further re-usage of the texture information for the remainder of the sequence of frames in a Group of Video Object Planes (GOV) inherently defines the candidate granularity for the reconfiguration (see also Section V).

We define a *reconfiguration* as the assignment of different resources that will be granted to and only to the job that is in a reconfiguration process. To this end, we distinguish *soft reconfiguration* by assigning new budget to the same resource and *hard reconfiguration* that requires migration of the job or part of the job to a new resource from the platform.

Taking into account the characteristics of our application domain (the coding aspects) and the complexity of hard reconfiguration (the recurrence time of reconfiguration), it is plausible to define the length of the GOV as a suitable candidate for the reconfiguration period [13]. At this granularity level, the negotiation on resources can take place and a new *guaranteed quality level Q* is assigned to a job. However, two disadvantages occur that will be discussed now.

A. Relatively long resource-reservation interval

For the decoding of arbitrary-shaped MPEG-4 video objects [5], the reservation of resources for the whole GOV requires that the system has sufficient resources for decoding each Video Object Plane (VOP). However, the MPEG-4 GOV length is not known in advance and is determined by the actual encoder. Therefore, the QoS control of the decoder has to decide on the reservation of resources for the decoding application for the whole length of a GOV. This GOV "frame set" has a variable length (the authors observed sequences of several hundreds of VOPs in one GOV). In the worst case, the decoder QoS control has to decide only on a fragment of the GOV size. Consequently, this approach can sometimes lead to a QoS decision for a lower quality level for a long sequence of VOPs. This lower quality level already occurs when only one VOP cannot be decoded within available resources.

B. Slow response on the increase of available resources

We have observed that the reservation-based QoS is also sloth in covering the increase of available resources. The time for the reallocation of resources and increase of the guaranteed quality level for an application is only possible at the end of a GOV. When the quality levels of other jobs change or when a termination of other applications occurs, these resources cannot be directly used for the subsequent VOP decoding. The decoding at a higher quality level starts at the first frame of the next GOV. In the case that such an increase of resources appears at the beginning of the GOV, the response time of the system might be too long for the system user. These two limitations motivated us to supplement the reservation-based model with a run-time QoS adaptation. The details of our approach are presented in Section VI.

V. Scalability of the AS-VOP Decoder

A. Arbitrary shaped MPEG-4 decoder

We have studied object-oriented MPEG-4 coding for the new proposed QoS concept. In MPEG-4, every Video Object (VO) is represented in several information layers, with the Video Object Plane (VOP) at the base layer. This VOP is a rectangular frame containing hierarchically lower units, called Macroblocks (MB). A group of VOPs forms is called a GOV.

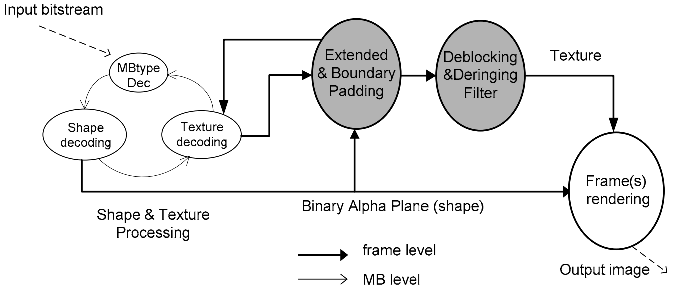

Fig. 3. Computation graph of the arbitrary-shaped MPEG-4 decoder. The shape and texture is processed at macroblock levels, the padding and postporcessing filters are at VOP level.

Scalability is becoming a key issue of future multimedia applications [14]. Depending on the application, a certain quality loss can be acceptable under circumstances as indicated in the following examples. The video decoding task can be pushed to a lower quality level with the aim to save some of the previously assigned resources for another application, like recording data from a surveillance monitoring system. A second example which is even more relevant for the domain of object-based video is the possibility to reproduce the video encoded at the highest level on a system with a lower level of the MPEG coding profile. In more detail, a bit stream encoded at Level 3 or 4 from the MPEG-4 Advanced Coding Efficiency (ACE) Profile may contain up to 32 video objects, as compared to Level 2 that is bounded to 16 VOs and Level 1 using only up to 4 VOs. If the number of objects composing the scene and their spatial resolution does exceed the limits of the system, less significant objects can be decoded at a decreased quality level, yet completing the original video scene with more information.

Figure 3 outlines a distributed version of the computation required for an arbitrary-shaped video object decoder. In our previous work [3], we presented a model for the shape-texture decoding at the finest granularity (MB level). This timing model is useful when the mapping requires usage of different processors even at the MB level. Here we present the graph for the complete decoding, which starts with parsing the bit stream syntax and coded data and ends with the completely reconstructed scene. The graph was simplified for presentation simplicity. The final visual scene can be composed of several VOs. In the presented graph, each task starts its execution when it has data on all inputs (as defined in Homogeneous Synchronous Dataflow Graphs).

In Figure 4 portrays the dynamic behavior of the size of arbitrary-shape video objects within a running video sequence. In specific cases of the figure, objects can grow between 1.5-5.5 times the initial sizes at the start of the sequence. This directly influences the requirements on resources.

The decoding starts with the *Shape & Texture Processing*, followed by *Extended & Boundary Padding* and the VOP decoding is completed by applying the *Deblocking & Deringing Filters* and providing the final shape and texture data to the *Frame Rendering*. The rendering is a shared task and composes the original scene from the video background sprite and several VOs superimposed on it. The background sprite decoder and QoS for it is not discussed in this paper; the reader is referred to [16] for more details. For the individual arbitrary-shaped VO, the complete task graph is instantiated. These independent instantiations output their results to the shared *Frame Rendering* task.

B. *Task-level scalability*

In order to deploy the combination of QoS techniques, we initiate the system with the worst-case mapping from the communication point-of-view, where we map each task to a different processor. This mapping is re-evaluated, to improve the mapping density within the *reconfiguration time*. Due to the complexity of reconfiguration, we consider only soft reconfiguration (as in Section IV) without a task migration.

Scalability of the coding by transmitting several streams containing different information layers was enabled already in the MPEG-2 standard and was used also for the design of systems with limited resources [17]. In our work, we targeted the identification of the decoding tasks with the option for saving a significant amount of (preferably) all types of resources. Second, the exited complexity of tasks limits the possibilities for incorporating an extra control and specialized types of processing. In our view, the balance in the resource utilization is a vital requirement of the mapping.

The most suitable solution is to keep the same level of the complexity and optimize the computation and communication resources at the same time by *enabling / disabling* tasks of the processing chain. This technique lowers all types of resources: computation and local storage of the target tile, where the task was planned to be executed as well as network connection streaming data to and from that task.

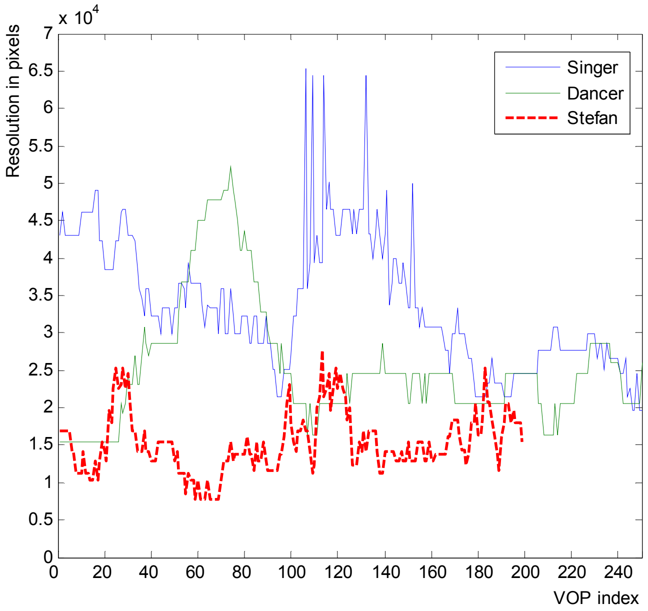

Fig. 4. Dynamism of the arbitrary-shape video objects sizes for sequences: *singer, dancer, Stefan* at CIF resolution with 25Hz frame rate.

We have defined the following three quality levels of our experimental AS MPEG-4 decoding.

- *Level 0* - Basic quality, the shape is fully decoded; the basic quality of texture after IDCT is communicated to the Rendering task
- *Level 1* - Medium quality; the MPEG-4 padding [2] of the texture data is activated, no artifacts on edges.
- *Level 2* - Highest quality; the complete chain with post-processing of de-blocking and de-ringing filters is executed.

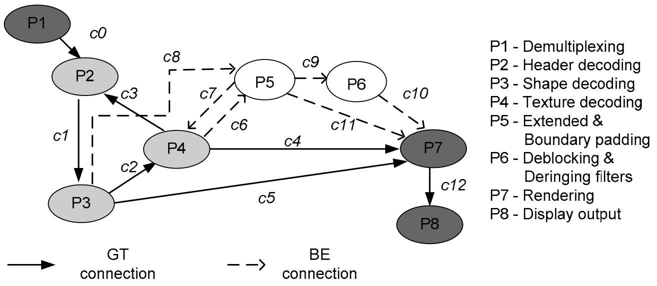

Fig. 5. The scalable computation graph of AS MPEG-4 decoder. Tasks P1, P7 and P8 are shared by all active instances of the AS VOP decoder. The kernel of shape-texture processing is formed by tasks P2-P6.

Our proposal for the scalable implementation is portrayed on Figure 5 with the indicated separation of shared tasks: *demultiplexing, rendering* and *output tasks.* Table I shows the distribution of task complexities in percentage of the computation resources for different video sequences.

VI. Predictable Computation with Best Effort Computation Based on Monitoring Services

The observation and analysis of ongoing computations in a system has received a lot of attention in literature. NoC monitoring systems [6] have been proposed in order to cope with observing the communication at run-time. This work was primarily driven by testing and debugging aspects. Passive hardware monitors make use of an industrial real-time solution for observing, called SPY [18]. We have decided to re-use the existing monitoring components for the run-time bandwidth steering and in combination with the Local QoS manager to enable *best-effort computing.*

TABLE I
Distribution of Task Complexity of AS VOP Decoding Tasks

Sequence	Shape & Text. decoding Q = 0	Padding tasks Q = 1	Deblocking & deranging Q = 2
Singer	72 %	17 %	11 %
Dancer	78 %	16 %	8 %
Fish	81 %	13 %	6 %
News	76 %	16 %	8 %
Average	76.75 %	15.5 %	8.25 %

TABLE II
Required Bandwidth of AS VOP Decoder at different quality Levels

Sequence	Shape & Text. decoding Q = 0	S&T decoding + padding tasks Q = 1	All tasks Q = 2
Singer	39.8 %	83 %	100 %
Dancer	39.4 %	81.6 %	100 %
Fish	37.8 %	83.5 %	100 %
News	42.6 %	89.3 %	100 %
Average	39,7 %	83.8 %	100 %

A. Monitoring enabled NoC architecture

Figure 6 illustrates the NoC architecture with routers (R) and Network Interfaces (NI). The NoC monitoring in Fig. 6 consists of configurable monitoring probes (P), attached to the R and NI components, and their associated programming model, and a monitoring traffic management strategy.

The *monitoring probes* are responsible for collecting the required information from the NoC components. The probes capture the monitored information in the form of events. Multiple classes of events can be generated by each probe, based on a predefined instance of an event model. Monitoring probes are not necessarily attached to all NoC components. The placement of probes is a design-time choice and is related to the cost versus observation-capability tradeoff.

The *traffic management* regulates the traffic from the Monitoring Service Access point (MSA) to the probes, which is required to configure the probes, while the traffic from the probes to the MSA is used to obtain the monitoring information from the NoC. Already available NoC communication services, e.g. guaranteed throughput (GT) or best-effort (BE) connections, or even dedicated solutions can be used for the traffic information for monitoring.

The above framework has been integrated in our experiments in the following way. The presented NoC with communication-monitoring features offers the combination of mixed GT and BE connections. The GT connections support the principles of reservation-based QoS control, while the BE connections fit to our QoS adaptation technique (presented in the subsequent section). The monitoring mechanism is needed to avoid non-optimal communication of data between tasks that will be completed after their deadline.

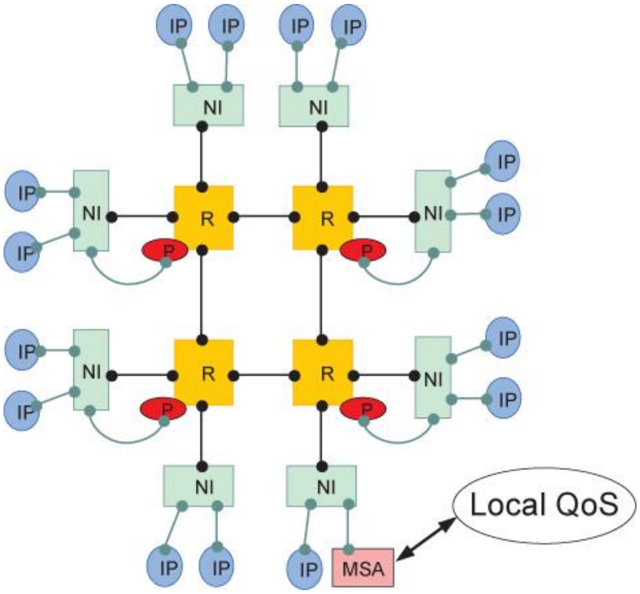

Fig. 6. The NoC architecture view with the Monitoring Service Access (MSA) connected to the Local QoS control manager. MSA is connected with probes (denoted by P) via guaranteed-throughput connections.

B. Resource-allocation model

We concentrate now on the long-term reservation of resources, which is based on predictable-computing principles and best-effort computation. Let us illustrate the resource-allocation model in our experimental setup. The mapping of tasks is following the worst-case mapping in communication, i.e. each task is mapped on a different processing tile. Our system architecture employs a 2x4 mesh Æthereal NoC with eight ARM processing cores. The ARM cores are one-to-one mapped to Network Interfaces (NI). We have implemented a centralized performance monitoring service. Each router was probed with performance monitors, which are able to monitor link utilization. Each monitor communicates performance data to the MSA by means of a low-bandwidth GT connection through the closest-located NI, which received an extra NI port for this purpose. The single MSA connects to NI (similar to Fig. 6) by means of an extra NI port.

Long-term reservation of resources. At the start of the GOV, the sub-task that estimates resource usage calculates the computation and communication resource requirements at all three quality levels. Next, the Global QoS selects the quality level at which all VOPs can be decoded. In our experimental setup (Fig. 7), Quality Level 0 will be selected because of the requirements of the VOPs with indexes from 13 to 22.

Best-effort computation. Our proposed solution is based on exploring the best-effort communication whenever it is possible. When compared to the GT connections, the BE connections do not have guarantees on the timing of data delivery. With the option to monitor the NoC connections, the Local QoS can verify at finer granularity (frame level) if there are available resources for BE communication. If the Local QoS received a positive response for all BE connections at a higher quality level, the extended computation at higher quality level is activated.

C. *Experiments and results*

In our setup, we have integrated alien traffic generators that program the system to a minimum level of communication activity. We have assigned the corresponding quality levels discussed in detail in Section V as follows:

- Level 0: c0–c5, c12;
- Level 1: all connections at Level 0 + c6, c7, c8, c11;
- Level 2: all connections at Level 1 + c9, c10.

The Local QoS has to monitor the connections c6–c11, as they are of the BE type. As is depicted in Figure 5, the initial quality is at Quality Level 0. Prior to starting the next VOP decoding, the Local QoS checks the status of the connections and if the estimated communication resources are available, then it activates the scalable tasks at the highest possible level.

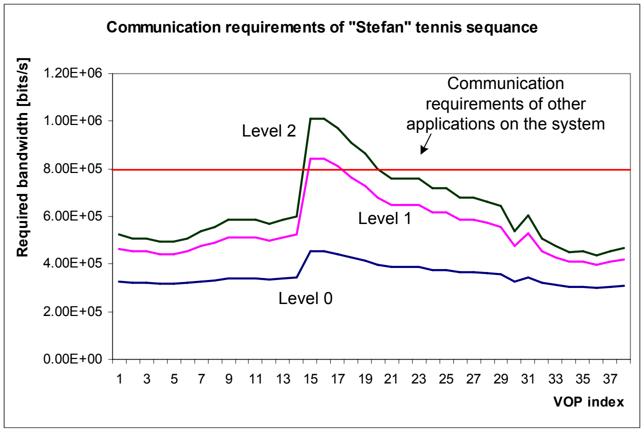

Fig. 7. Communication requirements of "Stefan" tennis sequence. The bold line represents the communication requirements of other applications also executed within the system.

With the novel mixed approach, the system is decoding only the VOPs 14–18 at Quality Level 0. Quality Level 1 is achieved for VOPs 13 and 19–22 and the rest of the GOV is chosen to be decoded at the highest quality level. The obtained improvement of quality is depicted in Figure 8. As it can be noticed, there is no improvement for frames that were decoded at Level 0. However, a significant improvement (to the absolute PSNR of approximately 35 dB) appears for the remainder of the VOPs.

The communication monitoring typically introduces overhead that is orders of magnitude lower than the required bandwidth of the experimental multimedia application. It can be readily concluded that the NoC monitoring allows the run-time adaptation of the decoding process to a higher quality. It should be noted that the obtained time fraction of 76% where the quality levels are increased, is highly dependent on the video input data and the run-time status of the platform

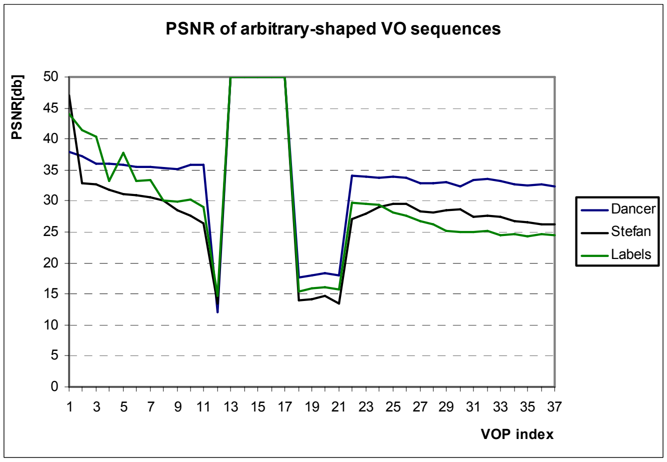

Fig. 8. Obtained PSNR for the arbitrary-shaped MPEG-4 video sequences. The "Stefan" tennis sequence has a resolution of 688×464 pixels, while the other two sequences are at CIF resolution.

VII. Conclusion

In this paper, we have presented task-level scalability for an arbitrary-shape MPEG-4 decoder. The scalable computation is essential for our hierarchical Quality-of-Service management. Object-based decoding shows rather dynamic resource requirements during the object life-time. Since the amount of objects is unknown in advance and the decoding characteristics are highly variable and chosen by the encoder, the guaranteed execution of all decoding tasks cannot be ensured. For this reason, we have proposed a new hierarchical QoS management system, featuring both intra and inter-application control.

We have employed a combined solution for reservation-based QoS management with run-time adaptation of the computation chain. This adaptation was implemented by using best-effort communication connections instead of the initialized guaranteed-throughput connections, where it was possible. The monitoring features in the NoC were formed by a Monitoring Service using run-time performance probes able to monitor link utilization, which were attached to all routers.

The complete system was experimentally verified with a network of eight ARM processor cores, executing an MPEG-4 Video Object decoder at the ACE profile and at CCIR-601 and CIF resolution. The proposed framework has shown that the adaptation at finer granularity, e.g. at the VOP level within a GOV, can improve the image quality significantly (experimental results show the absolute PSNR of approximately 35 dB with a quality improvement of 1–5 dB). Furthermore, it can be concluded that the monitoring of resources shortens the reaction time of the system to the system change due to video input changes or application changes.

The presented experiment on the combined QoS management highly depends on the monitoring features of an NoC. The timing issues of the reconfiguration were not taken into account because all connection are established and remain active or idle. Further research should focus on exploring these timing issues as well as on the "in-task" scalability issues.

APPENDIX

Figure 9 below presents in horizontal direction for each sequence the improvement of quality by activating the the padding, de-blocking and deranging tasks from the decoding chain. The original sequences are at CCIR-601 and CIF resolution.

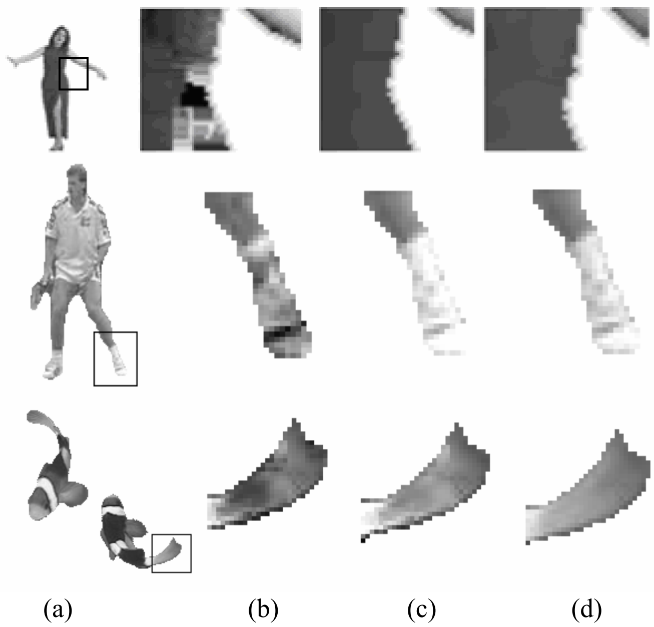

Fig. 9. Visual quality improvement by successive activation of computing tasks within MPEG-4 object decoding for three different objects. Column (a): original video objects; (b): basic decoding quality; (c): decoding with padding; (d): decoding with full post-processing.

ACKNOWLEDGMENT

The authors express their gratitude to Calin Ciordas and Kees Goossens from Philips Research Labs Eindhoven for their support in building the experimental setup and the helpful architectural discussion on Æthereal NoC.

REFERENCES

[1] J. Bormans, N.P. Ngoc, G. Deconinck, and G. Lafruit, "Terminal QoS: advanced resource management for cost-effective multimedia appliances in dynamic contexts", *Ambient intelligence: impact on embedded system design*, pp. 183–201, Kluwer Academic Publ., NL., 2003.

[2] ISO/IEC 14496-2:199/ Amd 1:2000, "Coding of Audio-Visual Objects- Part 2:Visual, Amendement 1: Visual Extensions", Maui, December, 1999.

[3] M. Pastrnak, P. Poplavko, P.H.N. de With, and D. Farin, "Data-flow timing models of dynamic multimedia applications for multiprocessor systems," *Proc. of 4th IEEE Int. Workshop System-on-Chip for Real-Time Applications (SoCRT)*, pp. 206-209, July 2004, Canada.

[4] R. J. Bril, Ch. Hentschel, E. F. M. Steffens, M. Gabrani, G. C. van Loo, and J. H. A. Gelissen, "Multimedia QoS in consumer terminals," *Proc. of IEEE Workshop on Signal Proc. Systems (SIPS)*, pp. 332–344, Sept. 2001, Belgium.

[5] M. Pastrnak, P. Poplavko, P. H. N. de With, and J. van Meerbergen, "Novel QoS model for mapping of MPEG-4 coding onto MP-NoC," *Proc. of 9th IEEE International Symposium on Consumer Electronics (ISCE)*, pp. 93–98, June 2005, Macau.

[6] C. Ciordas, T. Basten, A. Radulescu, K. Goossens, and Jef van Meerbergen, "An event-based monitoring service for Network-on-Chip," *ACM Transactions on Design Automation of Eletronic Systems*, Vol. 10, No. 4, pp. 702–723, October 2005.

[7] E. Rijpkema, K. G. W. Goosens, A. Radulescu, J. Dielissen, J. van Meerbergen, P. Wielage, and E. Waterlander, "Trade offs in the design of a router with both guaranteed and best effort services for networks on chip," *Proc. of DATE'03*, pp. 350–355, Germany, March 2003.

[8] N. Bambha, V. Kianzad, M. Kahndelia, and S. S. Bhattacharyyan, "Intermediate representations for design automation of multiprocessor dsp systems", *Design Automation for Embedded Systems*. 2002, vol. 7, pp. 307.323, Kluwer Academic Publishers.

[9] J.H.A. Gelissen, "The ITEA project EUROPA, A Software Platform for Digital CE Appliances", *Proc. Int. Conf. Consumer Electronics*, pp. 22-23, Los Angeles, CA, June 2002.

[10] C. Khan, "Quality Adaptation in a Multisession Multimedia System: Model, Algorithms and Architecture", *Ph.D thesis*, University of Victoria, 1998.

[11] D. Xu, D. Wichadakul, and K. Nahrstedt, "Resource-Aware Configuration of Ubiquitous Multimedia Service", *Proc. of IEEE Int. Conf. on Multimedia and Expo 2000 (ICME 2000)*, pp. 851-854, July 2000.

[12] B. Li, "Agilos: A Middleware Control Architecture for Application-Aware Quality of Service Adaptations", *PhD thesis*, Department of Computer Science, University of Illinois at Urbana-Champaign, May 2000.

[13] M.Pastrnak, P.H.N. de With and J. van Meerbergen, "Realization of QoS Management Using Negotiation Algorithms for Multiprocessor NoC", *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1912-1915, May 2006, Greece.

[14] S. Mietens, P.H.N. de With and C. Hentschel, "Computational Complexity Scalable Motion Estimation for Mobile MPEG Encoding," *IEEE Transactions on Consumer Electronics*, pp. 281-291, Feb. 2004.

[15] N. Brady, "MPEG-4 Standardized Methods for the Compression of Arbitrarily Shaped Video Objects," *IEEE Transactions on Circuits and Systems for Video Technology*, Ser. 8, Vol. 9, pp. 1170–1189, December 1999.

[16] Milan Pastrnak, Dirk Farin, and Peter H. N. de With, "Adaptive decoding of MPEG-4 sprites for memory-constrained embedded systems," *Proc of 26th Symposium on Information Theory in the Benelux*, pp. 137-144, May 2005.

[17] M. Ghanbari, "Two-layer coding of video signals for VBR networks," *IEEE Selected Area Communications*, vol. 7, no. 5, pp. 771–781, June1989.

[18] B. Vermeulen, S. Oostdijk, and F. Bouwman, "Test and debug strategy of the PNX8525 nexperia digital video platform system chip," *Proc. of IEEE International Test Conference (ITC)*, 2001, pp. 121–131.

Milan Pastrnak received the M.Sc. degree in information systems from Zilina University, Slovak Republic, in 1999. In 2002, he completed the post-graduation designer's course in Software Technology of the Eindhoven University of Technology. He received PDEng degree in September 2002.

Currently, he is with TOPIC Embedded Systems, Best The Netherlands. He is a visiting guest at the Video Coding and Architectures group, University of Technology, Eindhoven and he closely cooperates with Philips Research Laboratories, Eindhoven, The Netherlands. He received a best paper award at the IEEE ISCE in 2006 for his work on QoS management on Multiprocessor NoC. He is focusing on the application SW-HW co-design, heterogeneous multiprocessor systems, multimedia coding, quality-of-service for multimedia systems and system-level design.

Peter H. N. de With (M'81-SM'97) received the M.Sc. degree in electrical engineering from the University of Technology, Eindhoven, The Netherlands, in 1984 and the Ph.D. degree from the University of Technology, Delft, The Netherlands, in 1992, for his work on video bit-rate reduction for recording applications.

He joined Philips Research Labs Eindhoven in 1984, where he became a member of the Magnetic Recording Systems Department. From 1985 to 1993, he was involved in several European projects on SDTV and HDTV recording. In this period, he contributed as a principal coding expert to the DV standardization for digital camcording. In 1994, he became a member of the TV Systems group at Philips Research Eindhoven, where he was leading the design of advanced programmable video architectures. In 1996, he became senior TV systems architect and in 1997, he was appointed as full professor at the University of Mannheim, Germany, at the faculty of Technical Computer Science. Since 2000, he is with LogicaCMG as a principal consultant and he is professor at the University of Technology Eindhoven, at the faculty of Electrical Engineering. He has written and co-authored numerous papers on video coding, architectures and their realization. In 1995, 2000 and 2004, he coauthored papers that received the IEEE CES Transactions Paper Award and SPIE paper awards. Dr. de With is a senior member of the IEEE, program committee member of the IEEE CES and ICIP, SPIE's VCIP and former chairman of the Benelux community for Information Theory, scientific advisor of the Dutch Imaging school ASCII, IEEE ISCE and board member of various working groups.

Jef Van Meerbergen (M'87-SM'92) received the electrical engineering and the Ph.D. degrees from the Katholieke Universiteit Leuven, Belgium, in 1975 and 1980, respectively.

In 1979, he joined the Philips Research Laboratories, Eindhoven, The Netherlands. He was engaged in the design of MOS digital circuits, domain-specific processors, and general-purpose digital signal processors. In 1985, he started working on application-driven highlevel synthesis. Initially, this work was targeted towards audio and telecom DSP applications. Later, the application domain shifted towards high-throughput applications. His current interests are in system-level design methods, heterogenous multiprocessor systems, and reconfigurable architectures. He is the Associate Editor of *Design Automation for Embedded Systems*. He is a part-time Professor at the Eindhoven University of Technology, Eindhoven.

Dr. van Meerbergen is a Philips Research Fellow. His Phideo paper received the Best Paper Award at the 1997 ED&TC conference.