Hierarchical QoS Concept for Multiprocessor System-on-chip

Milan Pastrnak[a,c], Peter Poplavko[b,c], Peter H. N. de With[a,c], Jef van Meerbergen[b,c]
LogicaCMG Nederland B.V.[a], Philips Research Labs Eindhoven[b]
Eindhoven University of Technology[c]
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
M.Pastrnak@tue.nl

I. Introduction

Resource-constrained systems require that Quality of Service(QoS) is an integral part of the system design. In this paper, we provide a new QoS concept to be used in a large Network-on-Chip(NoC) setup for complex multimedia applications. The trend in video consumer systems is to use a heterogenous flexible platform in which many tasks are executed in parallel. For cost efficiency reasons, the platform cost and therefore resources are bounded, so that quality control of applications is inevitable. QoS management for Systems-on-Chip (SoCs) has been extensively studied for e.g. MPEG-4 3D graphics, wavelet coding, etc. [1] [2]. The proposed QoS management approach computes the resoure utilization as an algebraic function of the quality settings, based on e.g. the number of graphical triangles to be processed. However, in our target application domain, the execution time also depends on the input data. The input data dependency is particularly important for advanced multimedia applications, where video content is partitioned into independent video objects instead of rectangular frames of samples. As objects can change over time, in size, shape and texture contents, the processing requirements are more variable then with conventional video. For this reason we have adopted the MPEG-4 standard as an example of object oriented video processing.

In this paper we consider a class of QoS systems that relies on predicting the execution times of the application at run-time taking the data dependency into account. We propose a control mechanism based on a linear prediction model that interacts with a higher level QoS management unit to adopt quality and resource settings for our application. In initial experiments with our new concept, we found that the savings in computation is at least in the order of several tenths of percents.

[1]Supported by the European Union via the Marie Curie Fellowship program under the project number HPMI-CT-2001-00150.

II. Problem Statement

In this paper we study a complex multimedia application that can have very dynamic execution time characteristics per frame. In our previous work we motivated the use of a multiprocessor system-on-chip (MP-SoC) as a target platform for such applications [3]. MP-SoC is a reconfigurable platform and by using multiple processors executing a plurality of tasks, the QoS management becomes a multi-dimensional problem. In this paper we concentrate on the problem of mapping MPEG-4 shape-texture decoding as an example of a complex multimedia application. The object-oriented video is a more dynamic application in terms of resource usage (computation, memory, communication) than with regular video signals. The above-mentioned dynamism of processing of video objects in combination with the multiprocessor target platform, forms a complex mapping and run-time control problem. We concentrate on mapping MPEG-4 decoding on a distributed set of processors with the aim to control the complete application with respect to execution flow and output quality levels. Preferably, the mapping is scalable in terms of QoS and introduces a model for execution control.

III. QoS Model for Applications on MP-NoC

We focus on a complex state-of-the-art multimedia application (MPEG-4 shape-texture decoding) that can have very dynamic execution time characteristics per frame, and of which several instantiations can be executed in parallel. In our previous work we motivated the use of a multiprocessor system-on-chip (MP-SoC) as a target platform for such advanced applications [3], which makes also QoS management a multi-dimensional problem.

The architecture of our QoS concept is based on two negotiating managers, instead of the conventional single resource manager. We distinguish the *Global QoS manager* that controls the total system performance involving all applications and the *Local QoS manager* that controls an individual application within the assigned resources (see Fig. 1). Since the responsibilities of both managers are essentially different, it becomes apparent that a protocol for negotiation beetwen these two managers will be needed.

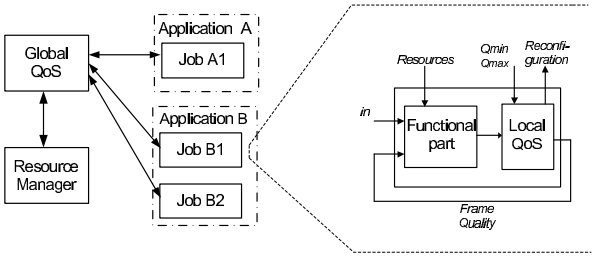

Fig. 1. Architecture of the proposed QoS.

Each application is divided in jobs and the platform should support each job for implementation, using the system resources and the software libraries. A job is split into different tasks, e.g. the MPEG-4 header parsing, the inverse quantization, IDCT, etc. In order to execute a job on a multiprocessor, tasks are mapped onto processors and other resources (memory, communication resources, etc.).

The use of the above architecture for system quality control makes sense if an accurate execution model for the application mapping can be provided. Our previous studies resulted in a quite accurate prediction model based on a linear parametrical equation at task level [3]. Nevertheless, not all of these parameters are *a priori* available when the task starts its execution. In [4], we proposed a prediction technique that estimates resource usage based on a set of parameters available from video headers and a complementary set of parameters from the decoding of previous frames. The resulting task level model offers continuous control of the decoding process with an acceptable accuracy, but at a finer granularity than current frame-based QoS systems.

Our mapping strategy exploits the predictability property of our architecture to enable a deterministic Quality-of-Service (QoS) from 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 resource distribution in case when each task is mapped to a different processor of the platform, and to keep independency between jobs.

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 *run-time refinement* of the resource usage based on the current resource-usage status of the system. Both phases are described in more detail below.

A. Off-line: intra-job mapping

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

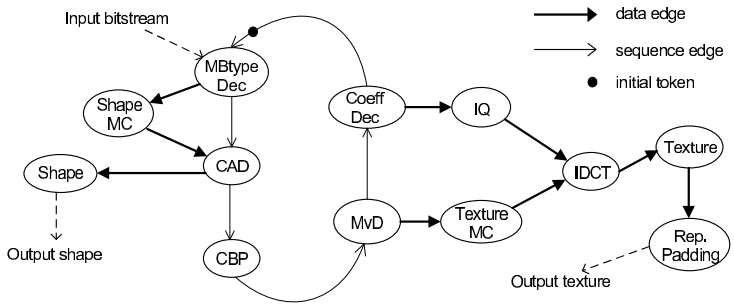

Fig. 2. Computation graph of distributed MPEG-4 arbitrary-shaped video-objects decoder.

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 settings should be adjusted continuously. The corresponding control system is called Local Quality Manager and is discussed in the Section IV.

B. Online: Resource management

The resource manager controls the available physical resources in conjunction with the Global quality manager, thereby using the operating points generated off-line. This works as follows. For a starting job, the Global quality manager chooses an initial quality setting by selecting an operating point. Based on off-line measurements of the anticipated quality Q, the manager strives for a quality setting that would 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 job has a high importance, the Global quality manager may decide to decrease the quality settings of some other jobs to release resources for the new job.

IV. QoS for Arbitrary Shaped Video Object Decoding

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). Figure 2 outlines a distributed version of a computation model for arbitrary shaped video object decoding at the finest granularity (MB level). In this graph, each task starts its execution when it has data on all inputs (as in Homogeneous Dataflow Graphs). Our application model also offers options to skip some of the tasks under certain conditions (e.g. repetitive padding is sometimes not needed).

Figure 1 portrays the Local QoS connected with a functional part of the actual Job. A runtime steering mechanism periodically observes the difference between the estimated resource requirements and the actually used resources. Based on the estimation error and input data characteristics, Local QoS handles two types of situations. First, short-term variations of the resource utilization may occur. To compensate for it, in the case that some of the tasks are scalable, the Local QoS manager

can change their local quality settings. If this step does not reduce resource needs sufficiently, it can disable some steps of the decoding process (e.g. a deblocking filter). Second, when the change in required resources has a long-term nature that needs control, e.g. the size of an object changes significantly, Local QoS can request for reconfiguration and new reservation of resources to the Global QoS management unit.

To evaluate the performance of QoS management, performance is modeled into an objective function based on quality settings and deadline miss rate. If the Local QoS estimates a deadline miss for presentation, but the next frame depending on the decoded frame will meet the deadline, the decision unit of the Local-QoS manager executes the decoding and omits the presentation. The objective function for performance estimation is

$$f = 1-(M_r+S_r)-N_r+0.5N_{r1}+0.25N_{r2}..., \quad (1)$$

where M_r stands for the rate of frames that were decoded but never presented, S_r is the rate of frames skipped completely and N_r represents the rate of frames that were presented but not on time. We can extend our function with a more detailed focus on frames displayed two frames after expected (denoted as N_{r1}), or three frames (N_{r2}), etc. Another possible extension is to add the weighted effect of changing the quality of scalable tasks internally of the job. However, for the evaluation of our QoS concept, Equation (1) provides sufficient means.

V. Conclusions and Future Work

We have proposed a new QoS management system supporting the control part of a resource-constrained system. The QoS system is split in two layers: Global QoS for overall system control and Local QoS for single application control and resource management. Our control systems are based on input-dependent models for video execution. The models exploit a parametrical linear timing specification of task-based subapplications. This approach was implemented in an experimental MPEG-4 decoder for arbitrary-shaped video objects. We have observed that our proposed QoS technique saves considerable amount of unused resources (up to 64%) compared to conventional QoS based on frame skipping when a deadline is missed. It should be noticed that the amount of saved resources depends on the contents of the test sequence and the utilization of the system. We found that prediction of resource utilization can lead to significant quality improvement of the complete system when resources are offered for other applications. The proposed QoS mechanism runs fast enough to be executed in real time. In future, we intend to combine data-dependent application models with event-based applications.

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" in Ambient intelligence: impact on embedded system design*, pp. 183–201, Kluwer Academic Publisher, The Netherlands, 2003.

[2] R.J. Bril, *Real-time scheduling for media processing using conditionally guaranteed budgets*, PhD Thesis, Technische Universiteit Eindhoven, The Netherlands, September 2004.

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

[4] M. Pastrnak and P.H.N. de With, "Data storage exploration and bandwidth analysis for distributed mpeg-4 decoding," in *8th IEEE Int. Symp. Consumer Electronics (ISCE)*, 2004.