Improving Resource Utilization for MPEG Decoding in Embedded End-Devices

Michael Ditze **Peter Altenbernd** **Chris Loeser**

C-LAB
Fuerstenallee 11,
33098 Paderborn, Germany
Email: {michael.ditze,peter.altenbernd,christoph.loeser} @c-lab.de

Abstract

Video streaming applications (e.g. video conferencing, video-on-demand) increasingly strive for deployment in small embedded systems that traditionally exhibit small computational resources as well as low speed internal network connections, e.g. set top boxes, mobile phones or PDAs. Whereas the latter is addressed by the new scalability features in the MPEG-2 and MPEG-4 standards , the computational resources still must be used effectively, all the more as continuously improved compression algorithms increase the computational demands. In order to guarantee a frame-per-second rate that satisfies the requested Quality of Service (QoS), a real-time scheduling mechanism is required that accounts for the specific needs of the respective compression standards along with a corresponding mechanism for Admission Control. The introduction of different frame-types in MPEG that require varying resources for decoding and the possibility of variable-bit-rate encoding, however, result in strong workload imbalances and unpredictability that do not allow to exploit the available resources efficiently. Therefore, this paper presents a new approach that allows to balance the workload caused by MPEG stream decoding. It allows to timely decode an additional 16% of frames compared to traditional solutions. Furthermore, we introduce a method called Peak Notification that may reduce resource over-reservation by considerable 67% through workload peak prediction compared to common solutions. Both methods increase the QoS delivered to the client. To our knowledge, this is the first approach that balances the workload on a single processor to achieve better CPU utilization.

Keywords: MPEG-2, MPEG-4, Real Time System, Quality of Service, Workload Balancing.

1 Introduction

Multimedia applications like video conferencing or video-on-demand increasingly begin to cross the network domain border in a converging Internet-working world (Madsen et al. 2002). A good example for this trend is the development of the MPEG video compression standards. Originally and exclusively targeted at multimedia applications to be executed on workstations with abundant computational decoding power and LAN connections that provide a bandwidth of up to 10 Mbit/sec, this standard in its version MPEG-4 has been further developed to even suit application scenarios in embedded MANs (Mobile Area Network) that feature very restrictive bandwidth (less than 1 Mbit/sec) and limited computational power (Puri et al. 1998).

Whereas the restricted bandwidth is addressed by the improved scalability features in MPEG-2 and MPEG-4 that allow video streams to be transmitted with adjusted quality through heterogeneous networks with varying bandwidth, the restricted computational resources of the end-client still remain a bottleneck. This drawback is further reinforced by constantly improved algorithms for video compression that reduce the required bandwidth for video transmission, but also raise the computational burden for the decoding process. Especially when the client shall display video streams with a pre-defined Quality of Service (QoS) (Vogel et al. 1995), the available resources play a restricting role. QoS is hereby referred to as a collective measure of the level of service delivered to the client. It can be characterized by several basic performance criteria that apply to the transmission media as well as to the end-client. With regard to MPEG streams, these criteria may include a frame-per-second (fps) rate of 30 fps conforming to the NTSC standard, a superior image quality and a short end-to-end delay from the sender to the receiver.

Real Time Scheduling and Admission Control These requirements are hard to fulfill with regard to Embedded Systems that feature sparsely available computing resources (The MPEG-4 standard provides profiles applicable to resource restricted environments (ISO 1998), e.g. mobile phones or PDAs). A *real-time* scheduling policy is hence required to meet the timing requirements as imposed by the fps-rate. Real-time scheduling distinguishes from traditional CPU scheduling in that results must be obtained within a given time-constraint. The job of the *Admission Control* is to analyze the predictable behavior of the real-time scheduling algorithm and determine in advance if a given task-set is feasible. A task-set is called feasible if all tasks in the set finish execution before their deadline even in worst-case conditions, i.e. when all tasks are released simultaneously and compete for CPU-resources (Burns et al. 1997). We proposed an *Admission Control Manager* (ACM) that assigns resources to every task in the feasible task-set in (Ditze et al. 2000). Thus, the ACM determines if an additional stream‘s request for resources can be granted. Otherwise, the stream is rejected. In case the resource requirements vary dramatically over time as it is often the case in MPEG stream decoding, the ACM must continuously re-invoke the AC in order to adjust the re-

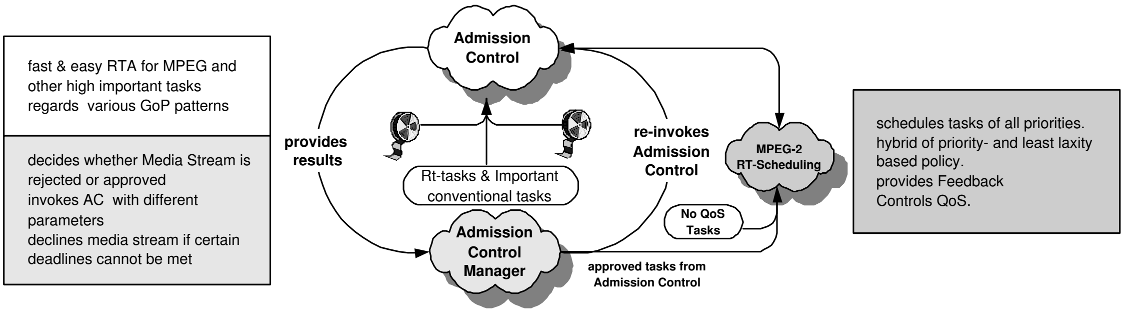

fast & easy RTA for MPEG and other high important tasks regards various GoP patterns
decides whether Media Stream is rejected or approved
invokes AC with different parameters
declines media stream if certain deadlines cannot be met
provides results
Admission Control
Rt-tasks & Important conventional tasks
Admission Control Manager
re-invokes Admission Control
MPEG-2 RT-Scheduling
No QoS Tasks
approved tasks from Admission Control
schedules tasks of all priorities.
hybrid of priority- and least laxity based policy.
provides Feedback
Controls QoS.

Figure 1: Collaboration of Real Time scheduler and Admission Control

source reservations. Fig.1 illustrates the cooperation among real-time scheduler, Admission Control and ACM. We presented such a method for AC based on a Response Time Analysis (RTA) along with an appropriate scheduling policy in (Ditze et al. 2000). RTA is a well known AC method derived from the Real Time Systems community.

AC itself is a time-critical task that competes for CPU-resources. Therefore, its processing must be fast and easy to qualify for continuous re-processing, but also accurate to avoid or at least decrease over-reservation. Since the accuracy of the analysis and the fast processing-time have contradictory demands, a tradeoff must be made. A possible such tradeoff is the *granularity* by which AC is processed.

In order to maintain the QoS, a corresponding method for Admission Control (AC) continuously re-processes in order to adjust to the frequently changing workloads and determine whether additional streams can be scheduled without reducing the QoS of other streams in the pipe.

Workload Balancing and Peak Notification Although an appropriate scheduling policy and AC guarantees that even in strong overloaded conditions a near to the optimal amount of MPEG frames can be scheduled, this amount may further increase if the strong workload imbalances in MPEG decoding can be smoothed. Similar observations derive from the networking world where traffic shaping models prove to increase resource utilization (Tanenbaum 1996) or in distributed and parallel systems where load balancing is necessary to effectively partition programming modules among processors. Whereas traffic shaping and load balancing are frequently used approaches in their environments, workload balancing on a single processor has been addressed very rarely.

The imbalances in MPEG in this cohesion either ascribe to the introduction of different frame types that exhibit different compression ratios or to the possibility of variable-bit-rate (VBR) encoding. They affect the decoding behavior and the predictability of the system in two ways:

The MPEG standards introduce three different frame types that exhibit different compression ratios and dynamic priorities for decoding that originate from decoding interdependencies. In case multiple high prioritized frames must be decoded simultaneously, as it is often the case when the CPU processes several streams, these frames easily consume the available computing resources. Consequently, succeeding lower-prioritized frames may be disregarded. As continues abrupt rate transition interferes the continues video flow and hence disturbs human perception (Steinmetz 1996), workload balancing is required to smoothen the adaption. Furthermore, averagely lowering the amplitude between maximum and minimum computation time may also result in more timely decoded frames. This reflects the impact of traffic shaping models that force traffic to conform to a predefined output rate (Tanenbaum 1996).

Therefore, this paper presents a new, simple method called *Workload Balancing* to re-distribute the resource requirements of multiple MPEG streams by adding decoding offsets, and thus putting streams in phase to one another. When combined with a scheduling policy designed for video streaming applications (Ditze et al. 2000), this may result in up to 44% more timely scheduled frames than in traditional solutions. Considered alone, Workload Balancing still may timely decode 16% more MPEG frames.

Whereas different compression ratios are responsible for the workload imbalances among frames that belong to *different* frame-types, the possibility of VBR encoding may result in strong varying resource requirements among frames of the *same* frame-type. These requirements may unpredictably diversify by a factor of two or even more. They therefore object the demand for a predictable environment with static Worst Case Execution Times (WCET) (Altenbernd et al. 2000) that allows the AC to precisely determine and allocate the required resources in advance. As the AC must allocate peak-rates that in case of sudden workload changes may result in strong over-reservation, the analysis must be continuously re-processed over subsets of the stream in order to decrease their impact. The granularity of the analysis thereby denotes the size of such a subset. Whereas a fine-grained analysis usually results in smaller over-reservation at the expenses of a high computational overhead, coarse-grained granularity increases the impact of workload changes. Therefore we present as a tradeoff a method called *Peak Notification* that informs the AC on sudden workload changes and allows to adjust the granularity accordingly. Both methods are applicable to MPEG-2 and MPEG-4 stream decoding.

The rest of the paper is organized as follows: Section 2 gives a short introduction to the MPEG standards. A summary on related work in relevant research areas is presented in Section 3. Section 4 introduces a method for Workload Balancing in MPEG streams, whereas Peak Notification is presented in Section 5. Section 6 then evaluates the superiority of the new approach, and finally, Section 7 gives outlooks on future work.

2 Introduction to the MPEG Standard

The MPEG standards developed by the Motion Pictures Experts Group have grown to become a worldwide standard for video compression reducing the workload on processors and networks by exploiting the intrinsic redundancy between consecutive video pictures. MPEG-4 covers a wide area of bit-rate ranges from below 64 kbits/sec for applications with extremely low bandwidth up to 4 Mbit/sec for video streaming applications (ISO 1998). As the allocated encoding bit-rate in MPEG-4 is not fixed, it may be further increased.

In contrast to its predecessors, MPEG-4 allows for the decomposition of video scenes into single audio-visual objects thus guaranteeing a high degree of user-interactivity. Each object can be separately encoded and transmitted in one or several Elementary Streams (ES). The improved spatial and temporal scalability features thereby allow to send base information of an audio-visual object required for a minimum QoS in a base ES, and further information improving the stream resolution or fps-rate in additional enhanced ESs.

In order to exploit the redundancy in video streams, MPEG-4 defines three particular types of Video Object Planes (VOP) that are temporal instances of an audio-visual object. These VoPs exhibit different compression ratios and are referred to as *I(ntrapicture)*-VOPs, *P(redicted picture)*-VOPs and *B(idirectional* predicted picture)- VOPs.

I-VOPs serve as reference VOPs to P-and B-VOPs whereas P-VOPs are predicted VOPs that collect relevant information encoded in former I-VOPs. They also serve as reference VOPs to B-VOPs. Consequently, I-VOPs and P-VOPs are also referred to as *reference VOPs*. B-VOPs can be either forward or backward predicted and likewise exploit redundant information encoded in previous or subsequent VOPs. The decoding times of back-to-back VOPs can thereby vary by factor of five or even more.

Since B-VOPs can be either forward-, backward-predicted or a combination of both, the MPEG standard distinguishes the order in which VOPs are encoded (*Display Order*) and the order in which they are transmitted (*Transmission Order*). Fig.2 further illustrates the interdependencies among the different VOP-types. The reference VOP a particular VOP relies on for decoding is denoted by the shaded boxes on the top right and the arrows pointing to that VOP.

A *Group of VOPs* (GOV) is a sequence of VOPs ranging from one I-VOP to the next. It complies with Groups of Pictures in MPEG-2. Even if MPEG does not standardize the GOV pattern, numerous streams often show the same fixed sequence. While *fixed spacing* and/or the use of GOVs is not required by the standard, it is so widely used, that a pair of parameters describes the spacing between I-VOPs and P-VOPs. The *N parameter* denotes the number of pictures from one I-VOP to the next, whereas the *M parameter* is used to describe the spacing between successive reference VOPs. However, this is not always the case. As each GOV may be self-contained, it is independent of others which allows for decoding without any knowledge about other groups. As I-VOPs serve as reference VOPs for the whole GOV, no further VOP decodes without having decoded the I-VOP first. Moreover, decoding a B-VOP requires to decode the respective I- or P-VOP first.

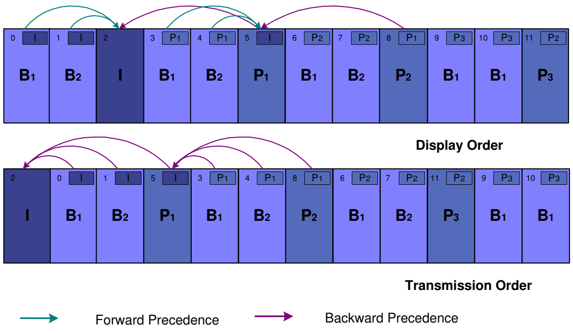


Figure 2: Transmission and Display Order in a GOV

MPEG-4 can be encoded in constant bit-rate (CBR) or variable bit-rate (VBR). Whereas CBR encodes every VOP at the same fixed bit-rate, VBR allows the bit-rate to vary, and hence ensures the same steady picture quality even in scenes that are hard to encode. As VBR reaches better compression-rates, we proceed on the assumption that MPEG-4 streams are encoded likewise.

3 Related Work

Most of the work that has been done in the field of workload balancing relates to traffic shaping issues in computer networks and load balancing in parallel and distributed systems.

Traffic shaping forces the network traffic to conform to a ceratin specified behaviour by implementing a specific policy that alters the way in which data is queued for transmission. Thus, the access to bandwidth can be controlled. Also, traffic can be prioritized and thus gets predictable. Traffic shaping is known to significantly increase resource utilization (Tanenbaum 1996). It is mainly applied in ATM networks. Some typical models for traffic shaping are the simple leaky bucket (Turner 1986), the token bucket and the token bucket with leaky bucket models.

The simple leaky bucket model collects incoming traffic in a leaky bucket from where it is drained with a constant rate *r*. In case the bucket size *s* is exceeded, further data packets are ignored. Thus, bursty traffic is shaped to conform to a constant rate.

A more advanced traffic shaping scheme is the token bucket model. Here, a token bucket sized *s* is filled up with tokens that arrive at a constant data-rate *r*. Each token represents a valid ticket that a sender can use to transmit one unit of data. Likewise, incoming data units are stored in a buffer. One token in the token bucket accounts for one data-unit in the buffer. Once the data-unit is sent out the token is removed from the bucket. If there are more data-units in the buffer than tokens in the bucket, the sender must wait for the bucket to be filled up with the corresponding amount of tokens. In contrast to the leaky bucket model, traffic bursts limited by b are permitted to pass whereas the traffic does not exceed data rate r.

A combined method compensates for huge bursts that still can be transmitted in case of a large sized token bucket which may be undesirable. Here, the traffic that passed the token bucket feeds another leaky bucket where the rate of the leaky bucket should be significantly higher than the rate of the token bucket in order to prevent congestions. Another version frequently used in ATM networks to cope with VBR traffic is the Dual Leaky bucket method that concatenates two single leaky buckets. The leaking rates correspond to the negotiated peak cell rate, the maximum cell rate at which the user will transmit, and the sustained cell rate that corresponds to the average rate, as measured over a long interval.

Whereas these methods prove well in ATM networks, they are not very well suited for employment to shape the CPU load in bursty real-time environments. The reason for this is that traffic shaping models are not aware of timing requirements and consequently retain packets to conform to the leaking rate even if deadlines must be met. ATM networks hence only give QoS guarantees on a pre-negotiated servicing contract and reserve the appropriate network resources on a fair-share manner by adjusting the bucket leaking rates accordingly. This is not reasonable for MPEG end-client decoding where resource requirements may vary dramatically.

With regard to CPU resources, load balancing has become a crucial factor in distributed and parallel systems. The aim of these algorithms in such environments is thereby to cost-effectively equalize the processor's workloads. The term cost here relates to network transmission delay that must especially be taken into account when dealing with real-time tasks. A number of these algorithms that depend on the application environment, (e.g. in case of task dependencies the object is to minimize the completion time rather than balancing the workload) does exist ((Ander 1987),(Iqbal et al. 1986),(Lu et al. 1986)).

Essentially, load balancing mechanisms distinguish between static and dynamic algorithms. Static approaches assume a priori knowledge of the execution time and communication patterns to decide about processor mappings before run-time. A review of work on static approaches is given in (ElKhatib et al. 1997). Dynamic methods balance the workload according to decisions made at run-time by migrating processes to other processors. The metrics for such decisions may comprise the processor utilization, the CPU queue length, the amount of signaled events in a cluster, the processor advance simulation rate or even the pure existence of task interdependencies.

However, all these algorithms are applicable to balance the workload in multiprocessor- or even distributed environments where the respective CPUs cooperatively work on a defined set of tasks. To our knowledge, this paper presents the first step to balance the workload on a single processor in order to achieve a better CPU utilization.

4 Workload Balancing

We assume an open environment where the encoder is under control and generates streams with GOV headers. While the standard does not require GOVs, they are so commonly used that the following methods work with almost all available streams. During our work we did not encounter a single stream originating from other sources that did not exhibit GOV headers.

Traffic shaping in networks and load balancing in distributed systems are frequently used methods that may increase the resource utilization significantly. The adaptation of these approaches to single processor systems, however, has been addressed very rarely. The evaluation results of this paper demonstrate that shaped workload imbalances in temporarily bursty environments like MPEG decoding positively influences the performance.

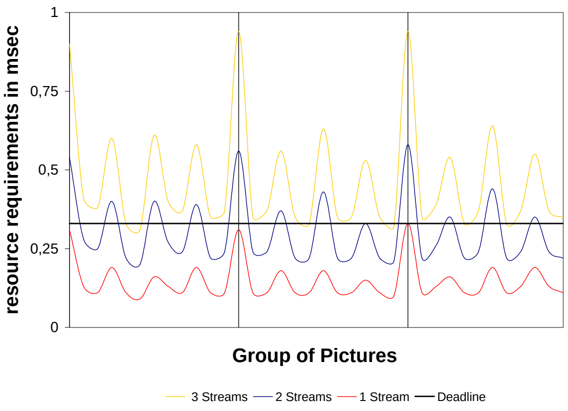
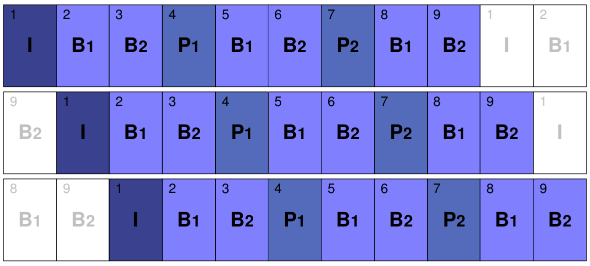


Figure 3: Workload Imbalances in MPEG decoding

The workload imbalances in MPEG-4 arise due to the structure of a GOV and the different compression ratios of VOPs. I-VOPs usually consume considerably more system resources for decoding than P-VOPs or B-VOPs. Consequently, the first part of a GOV typically requires abundant resources whereas further parts can be decoded in less time. This effect is further reinforced when multiple Elementary Streams that enter the system shall be decoded concurrently, e.g. as in video conferencng- or video surveillance applications. In case similar GOV patterns apply, several time intensive decoding tasks for reference VOP decoding may hit the CPU at the same time causing strong workload peaks. As a result, B-VOPs that follow reference VOPs in display may not be timely processed, thus causing a lower fps-rate and jittered display. Fig.3 illustrates the resource requirements of several typical stream scenarios broken down into GOVs and particular VOPs. The sample is an extraction of a MPEG-2 WCET analysis and comprises 3 GOVs only. Workload peaks occur when reference VOPs are decoded, where higher peaks represent I-VOPs decoding and lower peaks denote P-VOPs. The more streams enter the system, the higher the amplitude between reference VOP- and B-VOP decoding.

Workload Balancing tries to re-distribute the uneven resource requirements by putting multiple streams in phase to one another such as decoding tasks do not start at the same time. In case an appropriate phase-offset is chosen, the CPU does not need to process time consuming reference VOPs simultaneously. As a consequence, workload peaks are reduced. This is absolutely permissable for multiple audio-visual objects that belong to different video streams. Objects, however, that jointly compose the same scene may leak synchronization in case Elementary Streams are continuously decoded with separate offsets, thus disturbing human perception. On the other hand, small offsets, e.g. 33 or 40 ms that represent typical slacks between consecutive VOP decoding, generally do not influence the perception (Steinmetz 1996), all the more when they are restricted to small objects within the whole video scene. An additional workaround to avoid displaying unsynchronized video is to slightly increase the size of the video decoding buffer by the selected amount of offset VOPs. For videos encoded at data-rates of 1Mbit/sec, this averagely results in about 5 Kbyte additional storage space per VoP. The additional overhead is acceptable to current CE embedded devices.

Fig.4 shows how Workload Balancing works in case 3 streams shall be scheduled. In contrast to common solutions that would simultaneously schedule all I-VOPs, our method defers two streams by a certain phase-offset. This offset amounts to 1 in case of the second stream and to 2 for the third stream. As a result, reference VOPs do not concurrently compete for resources. Furthermore, evaluation results prove that Workload Balancing allows to timely decode more VOPs compared to common solutions (see Section 6).



1 I	2 B1	3 B2	4 P1	5 B1	6 B2	7 P2	8 B1	9 B2	1 I	2 B1
9 B2	1 I	2 B1	3 B2	4 P1	5 B1	6 B2	7 P2	8 B1	9 B2	1 I
8 B1	9 B2	1 I	2 B1	3 B2	4 P1	5 B1	6 B2	7 P2	8 B1	9 B2

Figure 4: Workload Balancing

5 Peak Notification

Since the resource requirements for decoding of the respective VOP-types tend to vary significantly, the AC must allocate a peak-rate over a certain time-window for every task in the task-set in order to guarantee that every task instance meets its deadline. The allocated peak-rate hereby corresponds to the maximum WCET of the respective task instances. The disadvantage of peak-rate allocation is that more resources may be accounted for than actually needed, thus causing over-reservation. Apparently, a GOV builds a reasonable fundamental unit for such a time-window. The granularity expresses the amount of GOVs that form a time-window.

Processing an analysis with a low granularity, i.e. over a small amount of GOVs, usually results in low over-reservation at the expenses of a high computational overhead. The size of the time-window is small and the accuracy of the analysis increases since the impact of workload peaks restricts to the size of the time-window. At the same time, the amount of time-windows to be considered by AC grows, and along the computational expenses. Increasing the granularity also expands the time-window which reduces the computational effort, but also decreases the accuracy. For every task, the maximum peak-rate must be allocated for the whole duration of the time-window. In case of sudden workload changes, the effect on resource over-reservation may be dramatic. This presents the dilemma: On the one hand, an accurate analysis is desired to decrease the degree of over-reservation. On the other hand, a low granularity increases the computational expenses of the AC.

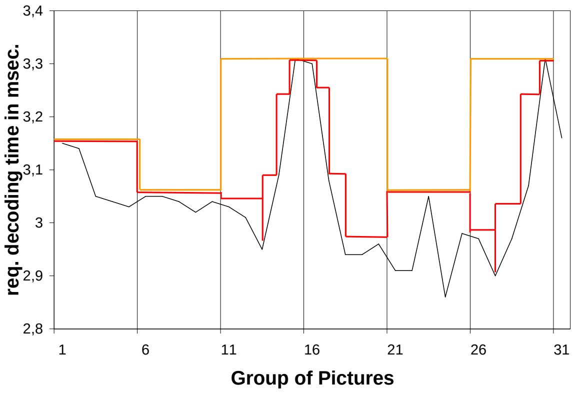


Figure 5: Impact of Peak Notification on Over-Reservation

The bright line in Fig.5 illustrates the amount of resources accounted for by AC in comparison to the real workload denoted by the black coloured graph. The AC is here processed with a granularity of 5 i.e. one AC instance is processed over 5 consecutive GOVs. The space between the two graphs denotes the amount of over-reservation. As expected, the over-reservation tends to be smaller when the workload stays relatively constant where it dramatically grows in case of sudden workload changes.

Peak Notification (PN) is a dynamic method that timely informs the ACM on sudden impending workload changes. The ACM may then dynamically re-adjust the granularity of the analysis, thus reducing the impact of workload peaks on over-reservation. The granularity may be decreased to the duration of the peak, or for better results, to one GOV per AC instance, i.e. a granularity of one. After the workload peak, the AC is processed with the former granularity.

PN may be easily realized in case the system components form an open system and hence are under control. The system may examine the video stream offline in advance either on the server- or the client-side, and detect potential *workload peaks*. A workload peak is hereby an intermediate resource requirement that exceeds the usual value by a certain, user-defined amount of resources. PN exploits the structure of MPEG streams and uses the user-data field in the header of a GOV to make notifications about the peak, its duration and the GOV-number where it occurs. An impact of PN is only ensured if the information is timely available to the AC. Consequently, the required information is stored in the header of a preceding GOV that exhibits a certain time-window towards the GOV that causes the workload peak. Optimally, this time-window has the size of the granularity used in the AC analysis. However, as the applied granularity is not known during the pre-analysis phase, a reasonable offset must be chosen.

Fig.5 illustrates the impact of PN on over-reservation. In contrast to traditional solutions that process AC with a static granularity, PN re-adjusts the granularity in case of workload peaks. The size of the time-window between two AC instances is intermediately decreased and the impact of the maximum WCET restricts to the duration of the smaller window. The traditional approach would reserve the maximum computation time over the whole time frame and thus dramatically increase its impact. The drawback of PN is the additional computational effort since reducing the time-window results in more AC instances to be processed. Please refer to the next section for further evaluation results on PN and its impact on over-reservation.

6 Evaluation

This section presents the evaluation results with regard to the impact of Workload Balancing and Peak Notification. Results were obtained from a benchmarking analysis tool that simulates scheduling and Admission Control at the end-client. Input values were taken from a WCET analysis of real MPEG streams

Experiment 1: In a first experiment, we examine the impact of Workload Balancing (WB) on the strong decoding workload imbalances in MPEG-2 streams. We select two scenarios where three streams and two streams, respectively must be decoded simultaneously and request computing resources as denoted by the WCET (Altenbernd et al. 2000) from the CPU.

In comparison to the common solution, WB significantly flattens the resource requirement line for the first scenario. Fig.6. shows the results: The amplitude between the resource requirement extremes decreases from 63 msec to 23 msec in contrast to the standard approach. Furthermore, the maximum CPU-time needed to decode all frames of an *ordered set*, i.e. all frames that share the same deadline (Ditze 2001) only amounts to 0.62 msec, and hence is reduced by more than 51% compared to the common solution. We made similar observations in the second scenario where two streams request system resources (Fig.7). Though the impact reduces, the amplitude is still decreased by 26% and the maximum resource requirement falls from 0.58 msec to 0.46 msec whereas the minimum resource requirement stays constant.

Experiment 2: In a second experiment, we evaluate the direct impact of WB on MPEG scheduling itself by combining WB with the scheduling policy Multimedia Least Laxity First (MLLF) that we developed for MPEG stream decoding (Ditze et al. 2000). We furthermore put the results in relation to two typical real-time scheduling approaches, a static and a dynamic one. Dynamic algorithms prove to be the most appropriate solutions for MPEG scheduling (Baiceanu et al. 1996).

The first one (Smart ELLF) is a smart version of the Least Laxity approach that avoids thrashing (Hildebrandt et al. 1999). The *laxity* of a task is defined as the maximum time a task can be delayed on its activation to complete within its deadline. The scheduler picks the task that has the least time available. The deployed version is smart in that it allows high-priority tasks like I- or P-VOPs to execute even if they have a negative laxity.

The second policy is a static priority-based driven policy that in contrast to MLLF assigns priorities to frames statically before run-time. It derives from the Rate Monotonic scheduling policy introduced in (Liu et al. 1973). The priority assignment relation P is $P(I) \geq P(P) \geq P(B)$ and consequently I-VOPs in the working set are preferred over P- and B-VOPs. VOPs that have passed their deadline are removed from the working set.

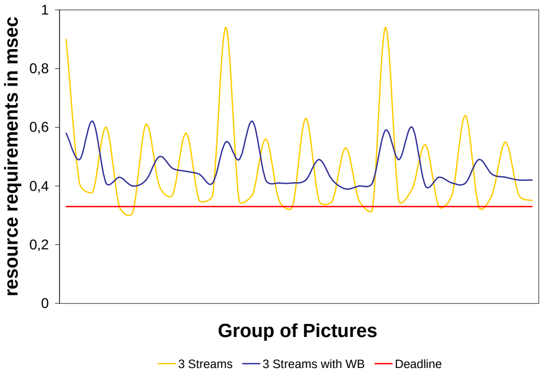


Figure 6: Workload Distribution for three streams using WB

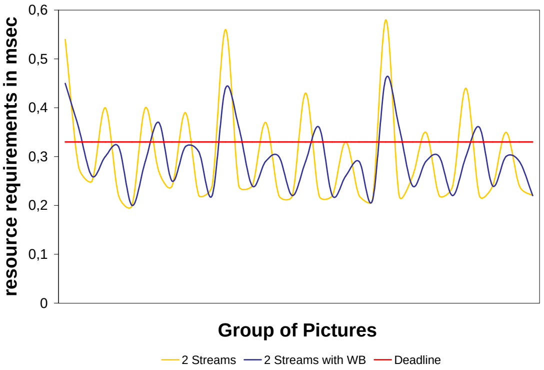


Figure 7: Workload Distribution for two streams using WB

Again, we chose two scenarios where 2 and 3 streams that cause strong overloads are scheduled. At first, we consider the 3-stream-strong-overload scenario (Fig.8). 2 out of 3 streams are put in phase by using a phase offset of 1 and 2, respectively. Compared to MLLF, WB again results in 16% more additional valid frames. The superiority of this new approach gets visible when compared to the common solutions. Here, 44 % more frames are timely scheduled than with priority-based scheduling, and with Smart ELLF, this value further increases to 62 %. At peak-rates, MLLF in combination with WB displays 64 % more valid frames than the next best approach. Furthermore, the new solution is still very insensitive to sudden workload changes and the number of valid frames almost stays constant between 21 and 24 during the whole sample.

We made similar observations in the second experiment (Fig.9), where the scheduler deals with 2 streams that cause strong overloads. This time, deploying WB results in 12 % more frames that meet their deadline, where 10-13 frames are constantly timely decoded. Compared to priority-based scheduling and Smart ELLF, the new approach decodes 32 % and 46% more valid frames, respectively.

Experiment 3: In a further experiment we show the superiority of Peak Notification. We therefore apply our AC (MLLF AC) with different granularties as introduced in (Ditze et al. 2000) to estimate the resource requirements and compare it to the multiframe approach for AC proposed by Nahrstedt (Kiwook et al. 1997). This approach reserves resources for each VOP-type by allocating respective WCET peak-rates.

At first we consider two streams whose impending resource requirements are estimated by an AC that uses a granularity of 2 (Fig.10), i.e. AC is processed once for every two consecutive GOVs. Whereas the Narstedt approach results in up to 37.5 % over-reservation at peak-rates, this peak decreases to 21 % with MLLF AC. On an average basis, Nahrstedt reserves 14.4 % more resources than required whereas MLLF AC degrades this amount to 10.2 %. Using MLLF AC in conjunction with Peak Notification further reduces the average over-reservation to just 7.3 %. As expected, the new approach is also very insensitive to sudden workload changes as denoted by the sketched line that represents the resource requirements. Here, we made similar observations for the second stream (Fig.11). MLLF AC reserves 20.5 % more resources than required. Peak Notification on the other hand just reserves slightly more than 10 % more resources. On an average, the new approach only wastes 6.8 % of resources compared to 9.9 % of resources that the Nahrstedt solution would require.

In the next experiment we set the granularity to 5 and consider the same two streams, expecting a stronger over-reservation (Fig.12). Whereas the over-reservation amounts to 32 % with MLLF AC, the Nahrstedt approach exceeds this amount by 8 %. Peak Notification, as expected, re-adjusts best to the sudden workload changes and only requires 22.5 % more resources than in fact needed. Considering the average, this amount reduces to 10.6 % in contrast to about 20 % with MLLF AC and 25 % with the Nahrstedt approach. Note, that the over-reservation peaks occur before the peaks described by the CPU-time requirements as AC is processed in advance.

The outputs of the second stream exhibit similar results (Fig.13). Whereas the amplitude between highest and lowest over-reservation ranges from 9 % to 43 % in the Nahrstedt approach, the Peak Notification considerably flattens the amplitude. Nahrstedt averagely over-reserves 21.8 % of resources, whereas this amount only increases slightly to 8.3 % with Peak Notification compared to the granularity analysis before.

Experiment 4: The next experiment processes AC over two streams in every granularity and determines the degree of over-reservation with respect to the ACs described earlier. The granularity amounts from one, where AC analyzes the impending resource requirements for every GOV, to a granularity that is processed over the whole stream. Figs. 13 and 14 illustrate the results of this experiment.

As expected, Peak Notification proves to be the most promising approach to reduce over-reservation. The over-estimation does not exceed the 13 % border that is reached when the granularity is set to 8. Furthermore, over-reservation stays constant once we chose a granularity greater than 13. This is understandable, because 13 represents the maximum amount of GOVs in between sudden workload changes for the first stream. Hence, we found the most coarse-grained granularity resolution. Further increasing the granularity does not have an impact as the granularity itself dynamically adjusts with regard to these workload peaks.

Furthermore, it seems to be surprising that none of the graphs ascends continuously increases monotonically. The reason for that is again the granularity itself. For example, consider a stream that has a workload peak at the seventh and the eight GOV. Applying AC with a granularity set to 7 will exactly split this workload peak and reserve the maximum computation times of the respective GOVs for two instances of AC, ranging from the 1^{st} till the 7^{th} GOV and from 8^{th} till the 14^{th} GOV. Consequently, the single workload-peak has an impact on the first 14 GoPs. Using a granularity of 8, however, causes the first AC instance to comprise the first eight GOVs including the whole workload-peak. Thus, the massively intense resource requirements of the workload peak only have impact on the first instance of AC whereas further instances are not concerned. This explains why a higher granularity sometimes may result in less over-reservation.

7 Summary

This paper presented new methods to improve resource utilization for MPEG decoding in embedded devices. At first, we introduced a model for Workload Balancing that successfully re-distributes workload imbalances in MPEG decoding of multiple video streams often found in video conferencing and video surveillance applications. As a consequence the display jitter that occurs during workload peaks and heavily disturbs the human perception decreases significantly. At the same time, Workload Balancing also has a positive impact on the amount of timely displayed video frames. Compared to traditional solutions, up to 62% more frames could be scheduled within their timing requirements. These results compose the employment of a new scheduling policy for MPEG streams that we presented in (Ditze et al. 2000).

Furthermore, we described a method called Peak Notification that informs the Admission Control on impending workload peaks. It allows to adjust the granularity of the Admission Control during workload peaks and thus may reduce the resource over-reservation by considerable 67% when combined with (Ditze et al. 2000). It represents a computational tradeoff between the two extreme situations where Admission Control is processed with a minimum granularity of just one GOV and a time-window that covers the whole video stream. The disadvantage is that it composes an additional computational overhead that can be neglected compared to the resources saved.

In future, both methods introduced here will be integrated in video clients that receive MPEG-4 streams through a RSVP and MPLS enabled Integrated Services network. This allows for evaluation in a typical workstation environment. Moreover, we will further examine the QoS services provided by the MPEG-4 standard in order to guarantee an end-to-end Quality of Service delivery, hence combining QoS mechanisms for the network and the end-client.

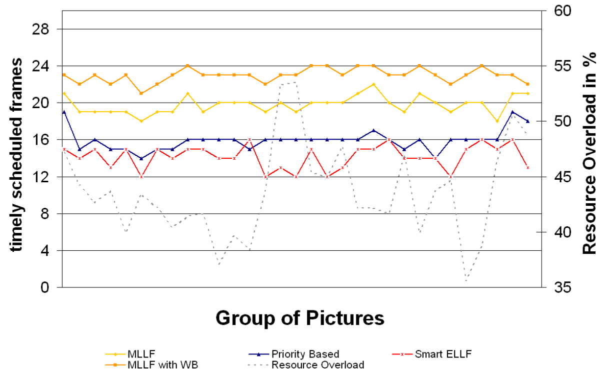


Figure 8: Scheduling three streams in heavily overloaded conditions with WB

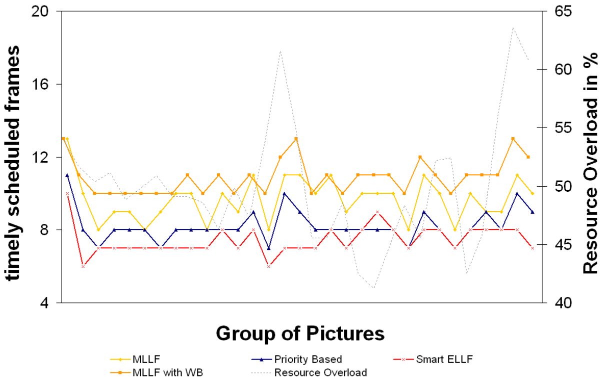


Figure 9: Scheduling two streams in heavily overloaded conditions with WB

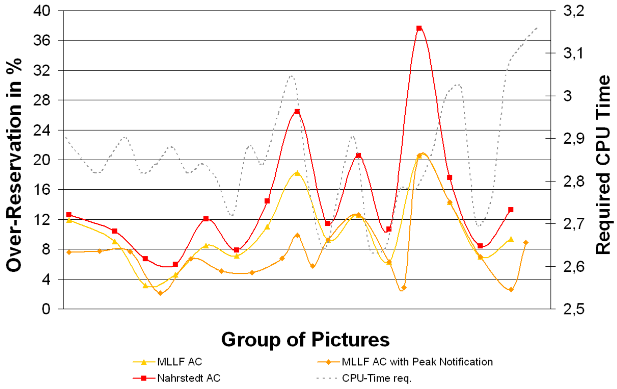


Figure 10: Resource Over-Reservation of the first stream with granularity=2

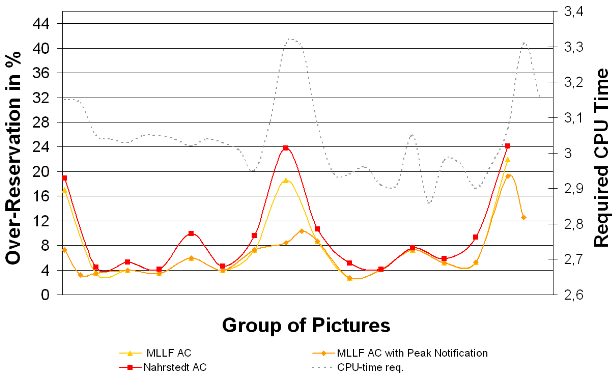


Figure 11: Resource Over-Reservation of the second stream with granularity=2

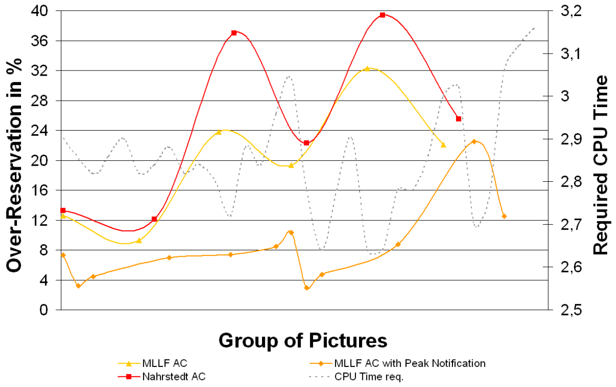


Figure 12: Resource Over-Reservation of the first stream with granularity=5

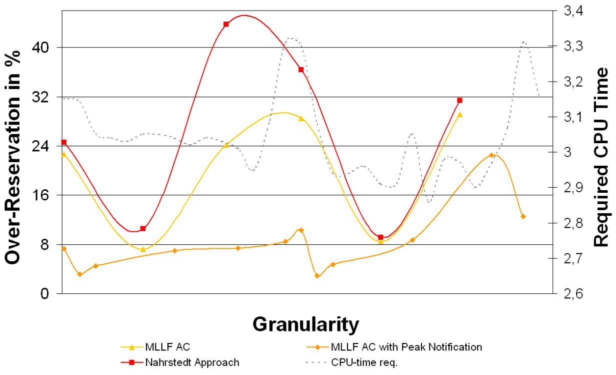


Figure 13: Resource Over-Reservation of the second stream with granularity=5

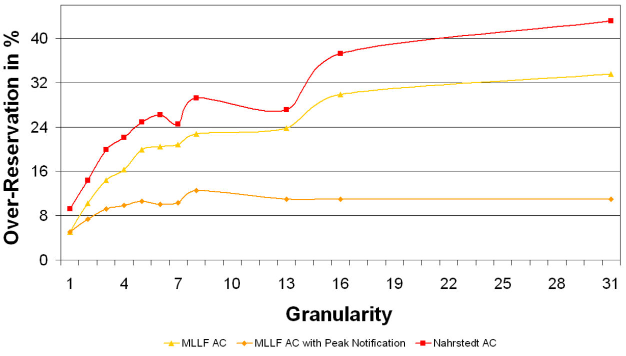


Figure 14: Granularity Analysis of the first stream

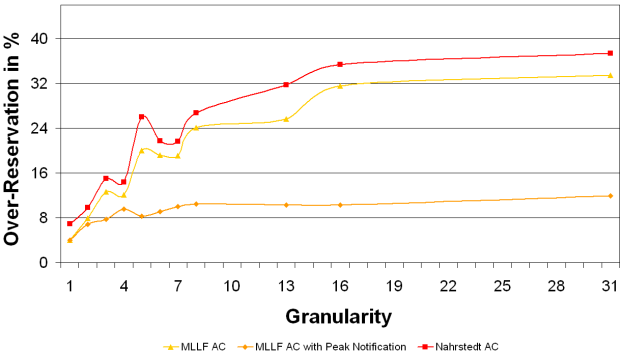


Figure 15: Granularity Analysis of the second stream

References

Altenbernd, P., Burchard, L., Stappert, F.(2000), 'Worst-Case Execution Times Analysis of MPEG-2-Decoding', *12th Euromicro Conference on Real Time Systems*, Stockholm, Sweden.

Ander, E.(1987), 'A Simulation of Dynamic Task Allocation in a Distributed Computer System', *Proceedings of the 1987 Winter Simulation Conference*pp. 768–776.

Baiceanu, V., Cowan, C.,McNamee, D.,Pu, C.,Walpole, J.(1996), 'Multimedia Applications Require Adaptive CPU Scheduling', *In Workshop on Resource Allocation Problems in Multimedia Systems, Washington D.C.*.

Burns, A., Wellings, A.(1997), 'Real-Time Systems and Programming Languages (second edition)', *Addison-Wesley, USA*.

Ditze, M., Altenbernd, P. (2000), 'A Method for Real-Time Scheduling and Admission Control of MPEG-2 Streams', *7th Australasian Conference on Parallel and Real-Time Systems*, Sydney, Australia.

Ditze, M.(2001), 'A New Method for the Real Time Scheduling and Admission Control of MPEG-2 Streams', *M.Sc. thesis, School of Computer Science, Paderborn University.*

ElKhatib, Khalil (1997), 'Dynamic Load Balancing for Clustered Time Warp', *M.Sc. thesis, School of Computer Science, McGill University*, Montreal, Canada.

Hildebrandt, J.,Golatowski, F.,Timmermann, D. (1999), 'Scheduling Coprocessor for Enhanced Least-Laxity-First Scheduling in Hard Real-Time Systems'.

Iqbal, A.M., Saltz, J.H., Bokhari, S.H. (1986), 'A Comparitive Analysis of Static and Dynamic Load Balancing Strategies', *Proceedings of the 1986 International Conference on Parallel Processing*pp.1040-1-47 Vienna.

International Oragnisation For Standardisation (1998), 'Information Technolgy -Generic Coding Of Audio-Visual Objects Part 2: Visual ', *ISO/IEC JTC1/SC29/WG11.*

International Oragnisation For Standardisation (1993), 'Coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbit/s', *ISO IEC JTC1/SC29.*

Liu, C., Layland, J. (1973), 'Scheduling Algorithms for Multiprogramming in a Hard Real Time Environment'. JACM, 20(1): pp.46-61.

Lu, H., Garey, M.J.(1986), 'Load-Balancing Task Allocation in Locally Distributed Computer Systems', *Proceesings of the 1986 International Conference on Parallel Processing*, pp.1037-1039.

Kiwook, K., Nahrstedt, K. (1997), 'QoS Translation and Admission Control for MPEG Video', *Proc. 5th International Workshop on Quality of Service (IWQOS'97)*, Columbia University, New York, USA, Pages 359-362.

Madsen, O.B., Nielsen, J.D., Schioler, H. (2002), 'Convergence', *1st International Workshop On Real Time LANS In The Internet Age*, Vienna, Austria.

Puri, A., Eleftheriadis, E.(1998), 'MPEG-4: An object-based multimedia conding standard supporting mobile applications', *Mobile Networks and Applications 3 (1998)*pp. 5–32.

Steinmetz, R. E.(1996), 'Human Perception of Jitter and Media Synchronization', *IEEE Journal on selected Areas in Communications, Vol.14, No.1 (1996)*pp. 61–72.

Tanenbaum, A.(1996), 'Computer Networks (third edition)', *Prentice Hall, USA.*

Turner, J.(1996), 'New Directions in Communication (or Which Way in the Information Age). ', *IEEE Coramunicatwns Magazine, Vol 24, 8-15.*.

Vogel, A., Kerherv, B., Bochmann, G., Gecsei, J.(1995), 'Quality of Service Management:a survey', *IEEE Journal of Multimedia Systems, Vol 2 no 2.*

1999.