ABSTRACT

This paper is motivated through the advantages offered by employing real-time (priority-driven) scheduling (RTS) for efficiently implementing quality-of-service guarantees. For employing RTS (e.g. for conferencing), a-priori knowledge about worst-case execution times (WCET) of the decoding process is required. With our new approach (focusing of MPEG 2 with variable bitrates) the resource usage compared to fixed-bandwidth allocation can be significantly decreased. Solutions for the WCET analysis are presented for two scenarios (Video-on-Demand (VoD), live-video). A significant reduction of overestimations during WCET analysis of the live scenario can be achieved by using our new technique. In the VoD scenario it is even possible to predict the exact execution times.

1. INTRODUCTION

Quality-of-service (QoS) and resource reservation play an important role for audio/video (AV) streaming. AV streaming applications are used for video-on-demand, TV broadcasting and live scenarios, such as video conferencing. An example for an AV streaming format is MPEG-2 [9], a popular and wide-spread standard for video and audio compression\(^1\). QoS guarantees allow to meet exactly the requirements for the playback of the delivered streams without loss of quality, e.g. packet loss or congestions during playback.

Many scheduling algorithms concerned with AV streaming do not consider timing aspects, which means some "best-effort" strategy is chosen without considering necessary and available computing and network resources or quality-of-service guarantees. As an alternative, fixed-bandwidth allocation or real-time scheduling allow a guaranteed QoS by taking properties of the streams into account. Compared to a fixed-bandwidth allocation, real-time scheduling has the advantage of a more efficient usage of available computing and network resources [11]. Real-time scheduling usually is known from the field of control systems, e.g. in avionics or automotive design. However, the opportunity to assign deadlines to a task makes real-time scheduling an interesting approach for multimedia applications where exact timing is required for e.g. smoothly displaying high-quality video streams such as MPEG-2.

Real-time schedulers allocate computing resource in a way that allows each process to meet its deadline by taking the worst-case execution times (WCET) into account. This means knowledge about the maximum time a process will run is required. In case the WCET is tightly estimated, an efficient resource utilization can be achieved. The WCET analysis deals with the problem of determining the maximum execution time without overestimating too much. Our experimental evaluations (see Section 3) have shown good prediction rates, resulting in an average resource utilization not below 85\% for each variable bitrate sample stream whereas fixed-bandwidth allocation never exceeded 39\%.

An efficient resource usage allows to implement cheaper designs and enable decoders to perform additional tasks, useful in the presence of multiple streams, like in conferences, or when using MPEG-4 technology which may contain several AV (streams within stream) or object descriptions such as 3D scenes. Our WCET analysis, however, is not restricted to be applied to real-time scheduling environments, as another opportunity is to implement a program which tests whether a given CPU is capable of decoding MPEG-2 streams.

The problem to overcome in order to achieve "good" WCET estimations, is to minimize the overestimation during WCET analysis. Especially, when using variable bitrates during encoding (for achieving the highest possible compression), the decoding times of the pictures in these streams have large differences, thus may result in enormous overestimations when using simple WCET techniques like [1, 8, 10]. In this document we show methods for two scenarios (video-on-demand and live-video) that do not limit the streams but try to reduce the gap between worst-case and the actual execution times. To the authors' knowledge this is the first paper addressing WCET analysis of MPEG

\(^1\)We concentrated on the MPEG-2 video standard although the presented implementations are general enough to cover also MPEG-1 streams.
In contrast to a simple method, that extracts the theoretically possible worst-case (thus ignoring all input data), we integrated the WCET analysis into the decoder (thus partly taking input data into account). This is achieved by two ways: Firstly, we analysed parts of the control flow of the decoding process, resulting in safe assumptions about relations between certain input data and the corresponding path taken in the control flow. Secondly, WCET information is collected in one early phase of the decoding, in order to make predictions about the execution time of a later phase.

The following sections deal with a VoD scenario solution (Section 2.1), and our new decoder for the live-video scenario (Section 2.2), respectively. Our results are shown in Section 3. Finally, Section 4 gives some conclusions.

2. WCET OF MPEG DECODING

A description of the MPEG video format is beyond the scope of this document, therefore we will concentrate on the description of the two scenarios.

2.1. WCET Analysis in a Video-on-Demand Scenario

An example for a VoD system could be a hotel application which provides access to videos on a server in each hotel room via television and settop-boxes. In the VoD scenario, streams can be examined and even modified before delivery. Hence, an exact WCET calculation can be done off-line and in advance. The only remaining problem is to group all available information in a reasonable way to be used by the end system’s scheduler.

The solution described here is based on precomputed execution times delivered within the MPEG stream and evaluated by the decoder in order to provide exact WCET information and deadlines for the scheduler on the client system. Having control over the streams on the server makes it possible to examine each stream, calculate the necessary computing time (with a decoder-like tool) and store this information as user data within the stream. User-data is defined by the MPEG standard in order to place additional data (i.e. data not related to the decoding process) within the stream. It can be stored in each group-of-pictures or picture header. In our case, we chose the picture header (Fig. 1), which allows to cover sequences of arbitrary length, e.g. information about a longer sequence of pictures can be stored in the user data field of each I picture.

The usual behaviour of an MPEG-2 decoder is to ignore user data, which means in case real-time scheduling is not required (e.g. a control screen might not need this functionality), any standard MPEG-2 decoder can be used to playback the modified streams without being influenced by the additional data. Although depending on the actual amount and type of WCET related data, the overhead caused by the additional user data in the stream is negligible.

2.2. WCET Analysis in a Live-Video Scenario

The strategy described in the previous section is not feasible for the live-video scenario, e.g. a video conference with several participants. The encoded videos are delivered online and a complete preanalysis is impossible.

Therefore, we restructured the usual decoding process in order to perform WCET estimations during the decoding process itself (see Fig. 2). The decoding process is organized in two phases, the first decodes all variable-length-codes (VLCs) and buffers the decoded data, and the second phase performs the remaining decoding steps. During a preprocessing step and the first phase WCET related information about the next phase is collected and finally delivered to the scheduler.

2.2.1. Preprocessing

The only relevant (w.r.t decoding times) VLCs are the encoded discrete cosine transform (DCT) coefficients. The preprocessing uses the length of a given picture, which can easily be determined, in order to estimate the maximum number of DCT coefficient VLCs, based on a comprehensive analysis of the control flow of the corresponding decoding code. Focusing on this figure, the WCET of the
first phase can be derived\footnote{The decoding time of all other VLCs is calculated using the theoretically possible worst-case decoding time.}. Although this WCET is only an estimate of the actual decoding time, the overestimation is rather small as shown in Section 3. For the reason of space limitations, refer to \cite{4} for a more detailed description of the preprocessing.

2.2.2. Phase One

During the first phase, all VLCs of a picture are being decoded. In addition to that, the WCET of the second phase is calculated and the decoded elements are buffered to be used during phase two. Moreover, the decoding time for the corresponding decoding step during phase can easily be identified based on available property descriptions. Therefore, at the end of phase one, the accurate WCET of the second decoding phase can be delivered to the real-time scheduler.

2.2.3. Phase Two

The second phase uses the decoded VLCs which were buffered during phase one to perform the rest of the necessary decoding steps. These include motion compensation and the inverse DCT (see \cite{9}).

The advantage of the two-phase approach is, that the overestimation of the WCET is limited to the first phase which is tightly estimated using the length of each encoded picture. However, the most time consuming parts of the decoding process take place in the second part of the decoding process, where our implementation provides the exact figures. The two-phase approach introduces some overhead to the decoding process (below 10\% for all of the sample streams), however, compared to fixed-bandwidth allocation this overhead is negligible.

3. EXPERIMENTAL EVALUATION

Since the VoD solution provides the exact execution times, the evaluation is focused on the two-phase approach for the live scenario, where overestimations of the execution time compared to the worst possible case could be significantly decreased. We tested the two-phase decoder with several MPEG-2 streams, represented by Fig. 3. It shows the overestimation together with the overhead required to decode each stream compared the results using fixed bandwidth allocation (giving the percentage of the actually necessary amount). The average overestimation plus the overhead for each of the sample streams was below 17\% resulting in a resource utilization of more than 85\% compared to less than 39\% using fixed-bandwidth allocation (see Fig. 4).

For the first time, the problem of worst-case execution times analysis of MPEG-decoding was addressed by this document, proposing solutions for different scenarios. Although only implemented for MPEG-2 Main Profile@Main Level, the two-phase decoder can be generalized to cover all possible MPEG-2 profiles and levels.

In the VoD scenario, our solution provides exact information about the execution time. For the live scenario, our two-phase approach significantly reduces the gap between the worst-case and the actual execution times. Moreover, the two-phase decoder does not rely on specially encoded streams making it a flexible tool in any environment.

Other applications of the results presented in this document include encoders able to add WCET related information into the user-data fields during the encoding process and a program that can determine whether a given CPU is capable of decoding MPEG-2 streams.
5. REFERENCES


