XVI IMEKO World Congress
Measurement - Supports Science - Improves Technology - Protects Environment ... and Provides Employment - Now and in the Future
Vienna, AUSTRIA, 2000, September 25-28

UNCERTAINTIES OF SCHEDULING SYSTEM FOR MULTIMEDIA

K. Nahrstedt and J. Bajcsy*
Department of Computer Science
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
*Department of Measurement
Slovak University of Technology, 812 19 Bratislava, Slovakia

Abstract: *Processing of multimedia applications in digital computing environments requires support of qualities, such as low jitter between video frames, known from current television or radio. This is especially difficult in these environments due to various noises coming from the hardware and software.*
In this paper we present (1) a novel scheduling software system for multimedia applications, (2) the measurement of quality parameters (e.g., skews, jitters) that are important for multimedia applications; and (3) evaluation of these measurements using uncertainties.
Our Dynamic Soft Real Time scheduling software system(DSRT) allows us to minimize the noises and to guarantee desired quality of service (QoS) for multimedia applications. We measure the quality of multimedia applications running under the classical time sharing scheduling system and the DSRT system. Evaluation of these measurements is performed using metrological evaluation methods. These methods allow us to quantify the improvements of the DSRT system using uncertainties and to optimize its performance.
Keywords: Multimedia Applications, Scheduling Software, Quality of Service, Metrological Evaluation, Measurement Uncertainties

1 INTRODUCTION

Processing of multimedia applications in digital computing environments requires support of qualities, such as low jitter between video frames, minimal synchronization skews between audio and video digital streams, known from the current television or radio. This is especially difficult in these environments due to various noises coming from the hardware and software.
We have developed a Dynamic Soft Real-Time scheduling software system (DSRT). We have measured its performance under noisy conditions. Important performance goal of this system is scheduling of a multimedia application under shared loads (noise) so that its quality parameters such as jitter and skew are preserved. We have evaluated the measured values using time diagrams and compared it with metrological methods of evaluation which use uncertainties. The analysis has shown that the latter methods allow us much better to optimize the overall performance of the DSRT system.

2 DYNAMIC SOFT REAL-TIME SCHEDULING SOFTWARE SYSTEM

DSRT is a novel scheduling software running on top of real-time extensions of an operating system (OS). The goal of the DSRT is to schedule multimedia applications so that (1) it provides low jitters and skews to the multimedia applications, and (2) it minimizes the noises which come from the hardware and software running concurrently with the measured multimedia application. The basic execution flow of DSRT, shown in Figure 1, consists of three phases. First, the considered multimedia application, which is a Soft-Real-Time (SRT) process, needs to go through a **probing phase**. During this phase, the SRT process runs in an isolated mode without any noises within a computer, the probing service measures its actual scheduling behavior of the application, and stores it in a profile. The profile is the input to the second phase **- the reservation phase**. This phase performs admission control to decide if the processor can schedule the SRT process, preserve its scheduling behavior requirement and concurrently share the processor with other application processes. If the SRT process can be admitted, processor binding contract is established, and the SRT process enters the third phase – **execution phase**. During this phase, the DSRT system schedules the SRT process together with other concurrent processes which are considered as the background noise. The DSRT scheduler together with the monitor, performance test and adaptation components guarantees that the multimedia application receives it desired quality parameters such as low jitter and minimal synchronization skew.

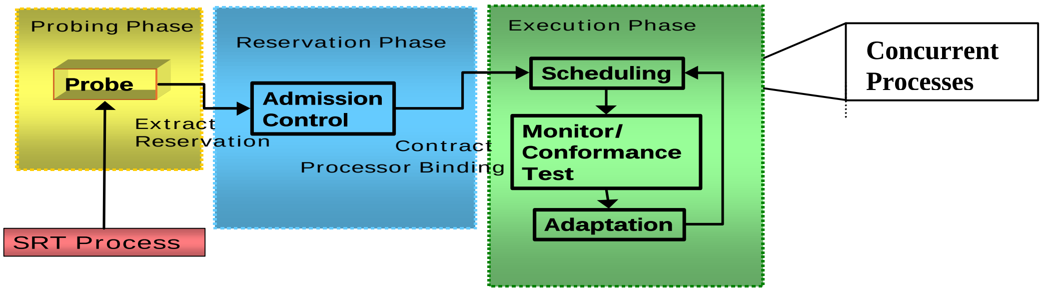
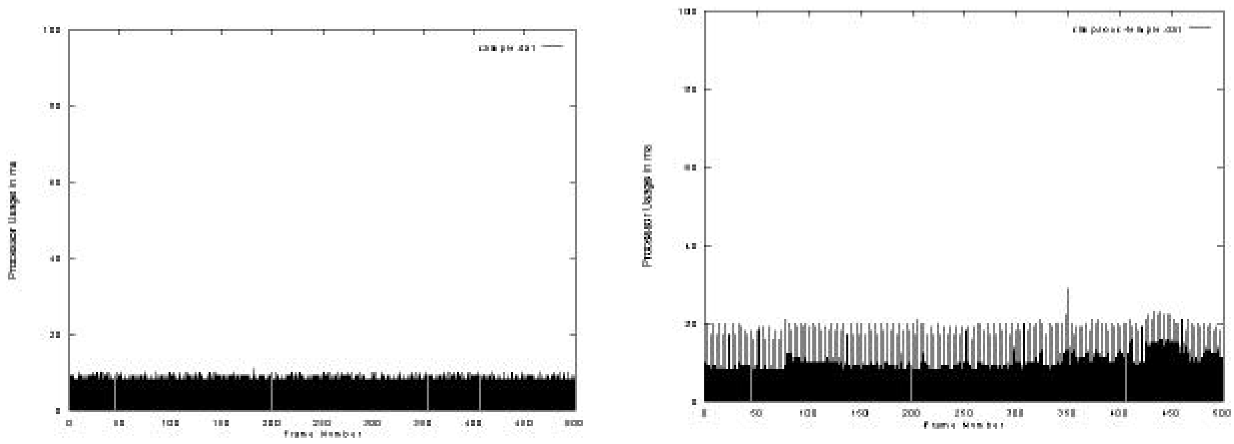


Figure 1. The basic execution flow of the DSRT scheduling software system.

Using these three phases, DSRT delivers soft processor time guarantees to a set of multimedia applications which fit the characteristics of CPU service classes. CPU service classes represent a specification interface which is used by applications (DSRT clients) to describe their Quality of Service (QoS) needs to the scheduling system. For multimedia application, we have designed two CPU service classes:

1. **PCPT (Periodic Constant Processing Time)** class allows specification of the *period (P)* for continuous media (e.g., period of digital uncompressed video is 33 ms if we want to support 30 frames per second video playback), and *peak processing time (PPT)*. A multimedia application can use this scheduling class if it needs no more than *PPT* CPU processing time every period *P*. Figure 2 (a) shows a processor usage pattern that exemplifies the PCPT service class.
2. **PVPT (Periodic Variable Processing Time)** class allows specification of the *period (P)*, *sustainable processing time (SPT), peak processing time (PPT*), and *burst tolerance (BT)*. A multimedia application can use this scheduling service class if it needs on average *SPT* amount of CPU processing time, but no more than *PPT* every period *P*. In addition, this kind of application may generate a processor usage burst in excess of *SPT*, but no more than *BT*. Figure 2(b) shows a processor usage pattern that exemplifies the PVPT service class.



(a) **(b)**

Figure 2. Video Frame-by frame processor usage (in ms) history for a PCPT class application in Figure 2(a) and for a PVPT class application in Figure 2(b). In Figure 2(a), the uncompressed video application performs a fixed amount of computation every 50 ms. The processing time per video frame stays approximately constant with *PPT* of 10 ms. In Figure 2(b), the video application plays MPEG compressed video stream at 10 frames per second. The *PPT* is 21 ms, *SPT* is 15 ms, and *BT* is 7ms.

2.1. Probing Phase

It is non-trivial for a multimedia application to determine the most suitable CPU service class and its usage parameters for reservation. This is further complicated by the fact that the processor usage may depend on input data, on the hardware platform and on the underlying operating systems. For example, running a software MPEG decoder on a SUN Ultra 2 computer is faster than on a SUN Sparc 10 machine. The DSRT system solves this problem by providing a **probing service.** A multimedia application enters the probing phase in which the DSRT system runs a few iterations of the multimedia application without any processor reservation and under lightly loaded CPU conditions. At the same time , the DSRT system monitors the client's processor usage iteration by iteration, and records them. At the end of the probing phase, the DSRT system analyses the processor usage history and derives a reservation matching the most suitable CPU service class and its parameters. The probing analysis extracts reservation parameters that satisfy two constraints: (1) the reservation must conform as closely as possible to the processor usage history of the application's software program collected during the probing phase, (2) the reservation should use minimum amount of processor resources. These two constraints ensure that the derived reservation does not over-estimate or underestimate its processor requirement. The probing analysis uses the *conformance test* [2] to extract and configure reservations satisfying these two constraints.

2.2. Reservation Phase

A multimedia application enters the reservation phase by submitting the configured reservation to the DSRT system. The DSRT system performs an *admission control test* (1) to check if there is enough processor resource to satisfy the guaranteed part of reservation, and (2) to determine which processor[1] to allocate this reservation to. The DSRT scheduler is based on the pre-emptive earliest deadline algorithm(EDF)[5] which must satisfy its schedulability condition. This schedulability condition drives the derivation of the admission control test. We assume that all multimedia application processes in the computer system are preempt-able and migrate-able to other processors. The admission control test must satisfy the following two conditions before accepting a new reservation request: (1) $S_i R_i$ £ RT, where R_i is the processor utilization of **i-th** application , and RT is overall processor bandwidth in the computer system available for all application reservations, (2) for each processor $j = 1,\dots,N$ $S_i R_i$ £ $RT_{j,}$ where RT_j is the processor bandwidth on processor j available for application reservation. The admission control checks if condition(1) is tru. If not, it rejects the request due to insufficient processor capacity in the multiprocessor system. Otherwise, it tries to locate a processor that has sufficient capacity to accommodate this new reservation request without changing processor binding of existing application processes. After the reservation request is accepted by the admission control test, it becomes a *reservation contract*.

2.3. Execution Phase

During the execution phase we need to enable different types of schedulers to enforce the soft-real-time reservations as well as guarantee that time-sharing applications do not starve. DRTS system deploys 4 different schedulers as follows. In order to support soft real-time guarantees for our CPU service classes, the DSRT system multiplexes the software real-time requirements of the CPU classes (PCPT and PVPT) into three different processor partitions, the *real-time partition*, *overrun partition* and *time-sharing partition*. Each of these partitions has its own scheduler and all these schedulers are scheduled by the *top-level scheduler*. The top-level scheduler uses a very simple credit/debit scheme to schedule the three partition schedulers. The processor time allocated to each partition scheduler is proportional to the relative size of its partition on each processor. The **real-time partitions** are dedicated to schedule the reserved (guaranteed part) runs of processor usage from multimedia applications, and they are managed by a centralised *RT Scheduler*[2], based on the EDF algorithm. The **overrun partitions** are dedicated to schedule the burst (statistically guaranteed part of a reservation) and overruns (non-guaranteed part of a reservation) processor usage from the multimedia applications, and they are managed by a centralised *Overrun Scheduler*. The **time-sharing partition** is dedicated to schedule only time-sharing(TS) processes so that traditional TS processes do not starve because of multimedia applications. The TS partition is managed by the underlying UNIX TS scheduler.

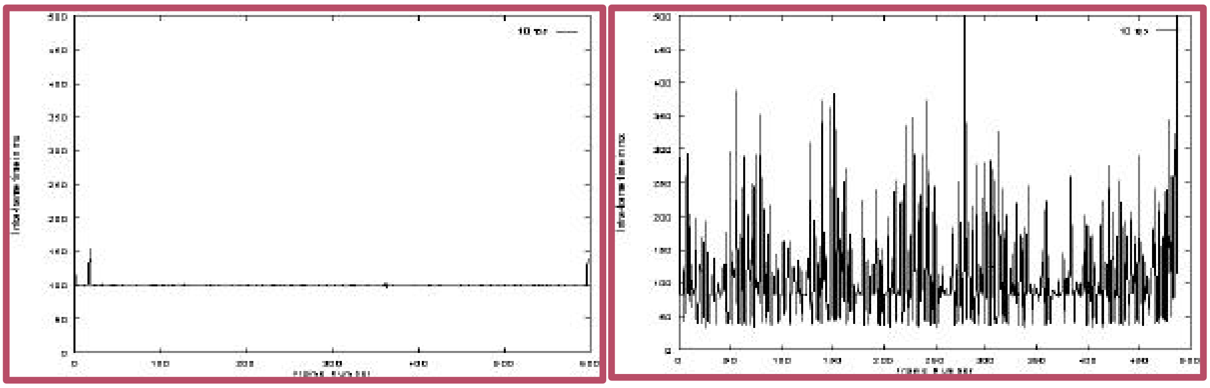
---

[1]We are considering a multiprocessor computer architecture such as the SUN Ultra SPARC 2 and higher Utra SPARC models.
[2] We have a centralised dispatcher on top of the SMP(Symmetric Multi-Processor) computer architecture which schedules the underlying SRT clients on different processors within the RT partitions.

Part of the execution phase is also a **system-initiated adaptation** provided by the DSRT system. Adaptation function of the DSRT system adjusts automatically QoS parameters in the reservation contract for a multimedia application, based on the application's actual processor usage time. What the multimedia application needs to do is to pick the most suitable adaptation strategy to adjust reservation contract. DSRT provides two different adaptation strategies: *exponential average strategy* and *statistical strategy* [2].

3 METHODS FOR CURRENT MEASUREMENTS OF QoS

We measure the quality parameter **- jitter** between consecutive video frames of a video stream during its playback via an MPEG video player application under noisy conditions (several background time-sharing processes). All experiments are performed on a Sun Utra 2 computer machine with two processors and 256 MB of memory, running Solaris 2.6. operating system. The time diagrams of jitter results under two scheduling software systems (see Figure 3) show that with our DSRT system we achieve an order of magnitute better results than with the classical FIFO (First In First Out) scheduling system of the general purpose operating system.



(a) **(b)**

Figure 3: Jitter quality parameter measurements between consecutive video frames. Figure 3(a) shows the jitter measurements under the classical FIFO scheduling system and the Figure 3(b) shows the jitter measurements under our DSRT scheduling system.

The evaluation methodology is done with help of time diagrams where we plot on the x axis the time in form of video frame numbers, and the y axis presents the measured jitter values per video frame in form of inter-frame times (in ms). The evaluation methodology is based on a visual evaluation and represents only a qualitative measurement of achieved system improvements. Based on this methodology we can visually see and determine if a system A performs better than a system B, but we cannot quantify more precisely the achieved improvements achieved by DSRT for multimedia applications.

4 EVALUATION of MEASUREMENTS VIA UNCERTAINTIES

We can quantify much more precisely the achieved improvements of the jitters for multimedia applications using methods for evaluation of uncertainties. These methods allow us numerically to express the measured results and their uncertainties. These quantified results are fed back to the DSRT system to improve on its adaptive behavior. Furthermore, based on these uncertainties, we find out which noise sources have the largest negative impact on the DSRT system behavior. We feedback this knowledge about the noise sources to the scheduling system and DSRT can use it for further optimization of its performance. We will elaborate on the evaluation methods using uncertainties.

Uncertainity of measured quality parameters comes from the insufficient knowledge of the value of measured parameters. The *sources of uncertainties* are sources which contribute to the uncertainties of measured parameters and therefore cause that the result of measurements can't be characterised by one value. The possible sources of uncertainities for multimedia quality measurements, done by our DSRT system, are as follows: (1) incomplete definition of measured parameter, (2) insufficient implementation of parameter definition, (3) non-representative set of measurements, (4) incomplete knowledge about the influencing conditions of the surrounding environment, (5) imprecision of etalon sampling frequency, and (6) changes when repeating measurements under the same conditions. We will examine the sources of uncertainities for multimedia QoS measurements as follows:

Definition of measured parameters: Basic sampling of any measurements within DSRT is done each 10 ms, i.e., T(us) = 10ms due to running the DSRT system within the user space of the computer system. This sampling period represents a time interval during which we don't have any information about the behavior of the measured quality parameter. This means that the measured parameter is discrete with the possibility of information losses within this interval. The frequency of sampling is f(us) = 1/0.01 = 100 Hz. With this exchange frequency of frames the human eye does not notice the change of video frames[3] . If we assume that the imprecision of source frequency is ± 1 Hz, this represents error within the sampling period ± 0.1ms. We can consider this value as an upper bound of the uniform distribution of density probabilities. Uncertainity of measurements is expressed as a standard deviation of arithmetic mean value of the uniform distribution u(us) = 0.1 : 1.73 = 0.058ms.

Implementation of parameter definition: Measured values in Figure 2 represent processor utilization times of individual video frames and measured values in Figure 3 represent inter-frame times between individual video frames. We can assume that the measurement of these times is derived from the sampling frequency, which means that the uncertainity of measurements will be the same as in the previous example, i.e., 0.058ms. Their statistical correlation is strong, i.e., their correlation coefficient is equal to 1.

Representative set of measurements: We measure the processing time of each frame, hence we have a complete statistical set. This means that the measured values are representative.

Knowledge about the OS environment: The conditions of the OS environment, i.e., non-controllable conditions in the operating system (e.g., I/O bus, context switching between DSRT system and multimedia applications, virtual memory and its impact on space allocation for the video application process in the RAM memory) are considered and encountered for within the DSRT system such that they do not bring additional errors to the quality measurements of multimedia applications. If this is not satisfied, there is not reason to compute uncertainties. The capability to differentiate time measurement can be derived from the frequency of hardware clock within the computer (in our experiment we used 200 MHz CPUs). This frequency is so high in comparison to the video application frequency (100 Hz) that we can discard the error from this source.

Precision of etalon sampling frequency: Etalon of sampling frequency can bring some uncertainity into the measurement. If its frequency is controlled by a crystal, the uncertainty coming from this source will be insignificant in comparison to the uncertainities coming from discrete measurements.

Changes during repeated measurements: Changes during repeated measurements under the same OS and background load conditions cannot occur, however this is in some circumstances difficult to achieve. The DSRT, as mentioned above, can encounter for OS uncontrollable conditions, however with the increase of Internet applications (e.g., web browsers, mailers), which very often run in parallel to multimedia applications and in asynchronous mode as background applications, slight changes in measurements can occur. We are currently in progress to estimate the errors contributed by these Internet applications and include them into the overall evaluation of measurements, done by the DSRT system.

The role of DSRT is to secure long enough time interval (reservation duration) for individual video frames. The derivation of the reservation times is based on an iterative approach which is first determined by the probing phase and then adapted to exponential average using history of data arrivals.

5 CONCLUSION

Analysis of sources for uncertainties shows that for qualitative functioning of applications, which use DSRT system, we need to choose an appropriate operating system (e.g., an OS which supports real-time priorities in order to enforce preemptions and real-time scheduling in the DSRT system), and monitor the quality of sampling to avoid jitters and skews for multimedia applications. The quality of parameters can be further improved if we compute during the adaptation of reservation times not only the *exponential average X* of processing times but also the *uncertainity* of measured processing times $U = k*u(A)$ where $k = 2$ and $u(A)$ is the uncertainity computed as a standard deviation of arithmetic

[3] Even frequency of 50 HZ is sufficient for a human eye without any loss in viewing quality.

mean value. Using this improvement we can expand on reservation times and decrease the number of overruns.

REFERENCES

[1] K. Nahrstedt, H. Chu, S. Narayan, QoS-aware Resource Management, *Journal on High-Speed Networking*, IOS Press, **7**(3) (1998), 227-255.
[2] H. Chu, K Nahrstedt, CPU Service Classes for Multimedia Applications*, Proceedings of the IEEE Multimedia Computing and Systems Conference*, (Florence, 8-12. June 1999), Florence. Italy, 1999, p.
[3] GUIDE to the Expression of Uncertainty in Measurement, *ISO, OIML, IEC, IUPAC, IUPAP, BIPM*, Paris, 1995.
[4] J. Bajcsy, Expression of Uncertainty in Measurement, *Journal on Electrical Engineering*, **48**(11-12) (1997), 328-332
[5] C.L. Liu, J.W. Layland, Scheduling Algorithms for Multiprogramming in a Hard Real Time Environment, *JSAM*, (1973),46-61

AUTHOR(S): Nahrstedt: Department of Computer Science, University of Illinois at Urbana-Champaign, 1304 W. Springfield Ave., Urbana, IL 61801, USA, Phone Int 217-244-6624, Fax Int 217-244-6869
E-mail: klara@cs.uiuc.edu
Bajcsy: Department of Measurement, Slovak University of Technology, Illkovicova 3, 812 19 Bratislava, Slovakia, Phone Int 421-7-60291313, Fax Int 421-7-60291600, E-Mail: julius@pluto.elf.stuba.sk