OS service update for an online adaptive embedded multimedia system

Kaïs Loukil, Nader Ben Amor, Mouna Ben Saïd, Mohamed Abid
CES Laboratory
ENIS National Engineering School
Sfax, Tunisia
Email: Kais_loukil@ieee.org

***Abstract*— The increasing popularity of nomad multimedia systems put new challenges for their design: increasing functionalities, limited energy and computation resources. These systems must provide a maximum application quality of service (QoS) in the presence of a dynamically varying environment (e.g. video streaming and multimedia conferencing) and multiple resources constraints. To respond to the changing resource availability and application demands, a new class of adaptation method is emerged. It combines the adaptation simultaneously upon the different layers related to the target system: hardware, application and OS. This paper presents an overview of a multilayer adaptation technique based on a global and a local manager. The global manager (GM) is used to handle large and long-term variations whereas the local manager (LM) is used to guarantee the real time constraint. This paper focuses on the LM that intervenes only in the application layer and OS layer.**

***Index Terms*— Adaptation, OS, Embedded systems, Quality of Service.**

I. INTRODUCTION

Profiting from scientific progress, in particular in the multi-media and telecommunication, the functionalities offered by the embedded systems became increasingly complex. They demand consequently increasing calculation capacities. This permanent rise in complexity is also derived from the permanent and fast evolution of the technological integration rate which gives the possibility to realize such complex systems. It increases by 50% each year, according to the law of Moore [26]. Thus, today it is possible to design a software/hardware heterogeneous system, single or multi-processor dedicated to the media domain (RISC, DSP, ASIP, IP) on the same chip: System-on-Chip (SoC).

The mobile nature of the majority of those systems pushes news design constraints. They have in general limited energy resources. They also involve in a dynamic environment and have to execute large variable workload. Those extern parameters have deep (and usually unpredictable) influence on the system performance. For those reasons, the use of the traditional design methods does not meet the desired performance as they focus only on the “static” applications parameters to generate adequate architecture [23]. New design methodologies have to be set up to address those recently introduced constraints especially energy consumption and the quality of service (QoS).

II. STATE OF THE ART

As the technology progress doesn’t yet propose new energy resources more efficient and always small and safe, the principal solution is the efficient energy consumption. Many works treated the energy consumption reduction for embedded systems at different level and using various approaches. High level power estimation [25] and optimization [22],[24] are examples of such studies. Some “low power” codesign tools were proposed [22].

All the previously described methods focus on “static” power estimation and reduction using application and architecture parameters. They don’t take on account “extern” system parameters such as data variation, the wireless network state, or the battery behavior impact on energy consumption and the system output QoS. The current tendency of the systems on chip design level is to generate reactive and on line adaptive architectures which for better satisfying the application constraints. These needs are expressed in terms of performance, QoS or energy consumption. They are fixed, on the one hand, following internal parameters such as the workload assigned to each application. To respect the real time constraint and on the other hand, they are dependant of the external parameters such as the residual energy in the battery and the user preferences (QoS and DDV).

Recently, there were many research contributions on the auto-adaptation for the autonomous systems. These adaptations can be applied to the following layers: the architectural layer, the operating system layer (OS) and the application layer. Hereafter, we will show these various techniques, their contributions and their limitations. Many HW adaptation techniques were proposed. One of them is the dynamic voltage scaling (DVS). It consists of adjusting the frequency and the power supply of the CPU. It is based on the application workload prediction using heuristic methods [4, 16] or worst cases CPU time estimation [2, 3]. HW adaptation was also applied to reconfigurable platforms. The change of the system architecture is made according to the needs for the application and the environmental constraints. Such method was applied for the partitioning of the system using two techniques: the former uses a heuristic algorithm [13] and the latter uses a genetic algorithm [15]. Many of researches have dealt with the operating system and middleware layer to provide predictable CPU allocation and adaptation services of the operating system [5, 7]. In [9, 12, 17, 1] the managers of CPU resources provide performances guarantees in soft real time. Schedulers further adapt the scheduling policy to handle the variations of application runtime [5, 6, 7]. In [1, 11, 19], the authors use a middleware layer to facilitate the adaptation of QoS for the application system which is submitted to the time constraints of execution and to the supplied energy. Several projects recommend the adaptation at the application layer for different purposes. For example, the authors of [18] explore the technique for adapting the behavior of the application to the constraints of energy consumption. Mesarina and others discuss in [8] how to reduce the energy for the "decoding MPEG" application using parameter modifications. In [12, 13, 14] two approaches are proposed for the deterioration of the quality of an object 3D to satisfy the constraints of resources and network bandwidth. Many researches have, recently, been elaborated to profit from the advantages of the previously described techniques and to reduce their restrictions. Most of these researches were based on methods which work on the various layers of the system such as model GRACE-1 [1] and TIMELY [20]. However, these approaches suppose that the material layer is not reconfigurable. The only possible adaptation to this level is on the CPU frequency.

III. PROPOSED APPROACH

Our model targets a mobile system with adaptive hardware architecture and running a set of parameter tuned applications (which are the case of the most poplar multimedia applications). It optimizes the system resources use under constraints of lifetime, real time and quality of service. The suggested model uses adaptation in the three layers hardware, operating system and application. For the hardware layer, adaptation is done using one of the different application implementations called configuration that are previously set up and characterized. Those configurations vary from a pure SW configuration (called config_sw) to a mixed implementation with several HW components (called config_hw). As config_sw involves only CPU resources, it will consume less energy than config_hw but will less performing. So those, the system can choose the adequate configuration according to the constraints and the user preferences. Adaptation at application layer is performed by modifying the application parameters or algorithm according to the imposed constraints. The third adaptation level is made at the operating system layer in order to allocate necessary time CPU for each task (we suppose that each task is a one application). This approach comprises primarily two managers (global manager and local manager). The global manager can intervene in the three layers in order to answer the great variations of the system constraints whereas the local manager intervenes only in the layers application and in the operating system. It is of a great importance to mention that the LM is set up to control the respect of the real time constraint of the system. The LM can also question the global manager if it does not manage to find a solution to solve the problem. In this paper we focus our studies on the Local manager. We describe here operation and the necessary steps to implement this technique. More details about the global operation of this approach exist in [10, 21].

IV. LOCAL MANAGER APROACH

Local manager "LM" intervenes at the application and the operating system layer (see figure 1) to satisfy real time constraint. It uses the "watch dog" technique, which permits to detect any overtaking in the expiry time of any task. Since the embedded multi-media systems are often subjected to hard real time constraints, if one of the tasks misses its deadline, the LM must check if an eventual overtaking influences the normal system function (i.e. if all the tasks are running normally throughout the hyper period of the system). If there is no influence, the LM does not intervene and the system continues to function with the same (application and architecture) parameters. In a case of deadline missing, firstly the LM tries to reduce task execution time (referenced as Texe) locally with a modification of application parameters. The choice of the new applications parameter is based on off-live established model (Texe=f(application parameters). When the LM finds the adequate applications parameters, it sends its instructions to both application and OS adapters to change their parameters. In the opposite case, the LM requests the GM to reconfigure the totality of the system using more complex mechanism such hardware reconfiguration. To respect the real time constraint it is necessary that each task finishes its execution before the end of its period. Since we work in a multitasking context, we assign for each task an amount of time, during which it finishes its execution, in order that the entire task can be carried out within its period. In addition to this portion of time assignment, we use the adequate scheduling algorithm to guarantee that the execution of each task finishes before the end of its period.

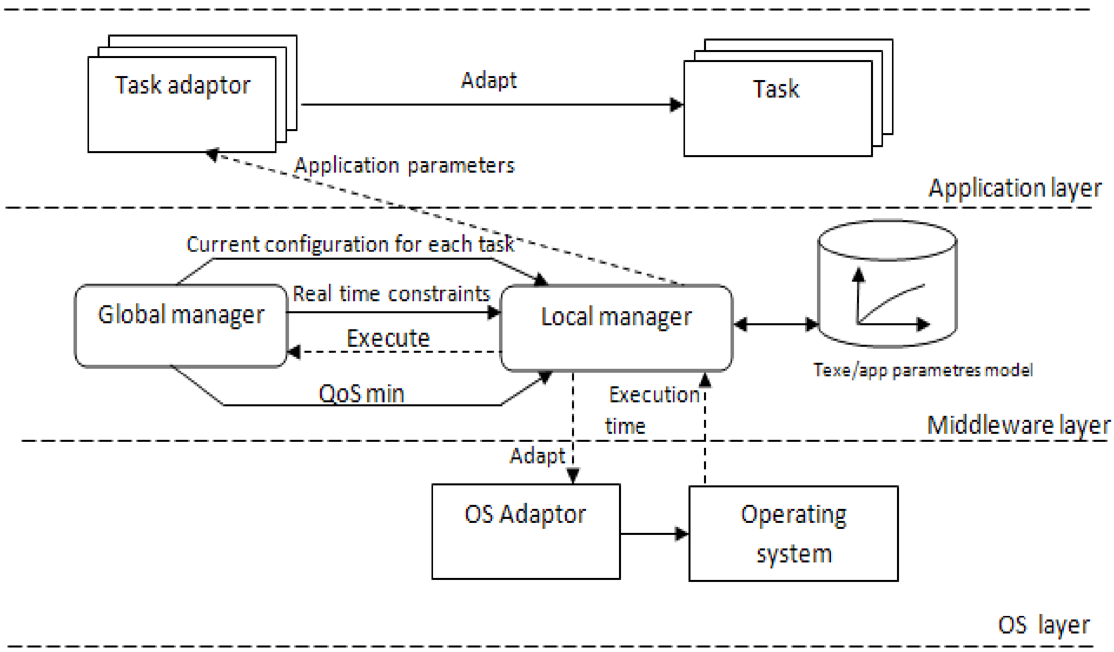

Figure 1. Local manager aproach

A. Allocation of the budgets of time

We suppose that our system executes a set of periodic tasks.

In order to respect the real time constraint, all the tasks must be running during their period. Thus, in the system hyper period "h" the task Ti of period pi will have to be running $ni = \frac{h}{Pi}$ times. Thus, the problem of assigning to each task Ti an execution time Tei is given by equation 1 where n is the number of applications.

$$\sum_{i=1}^{n} Tei * ni <= h. \qquad \text{E1}$$

Generally, we must satisfy the equation2:

$$\sum_{i=1}^{n} \frac{h}{Pi} * Tei <= h \qquad \text{E 2}$$

This work is operated by the global manager. In this way, all tasks can be running in the system hyper period. But this condition is not sufficient to guarantee the execution of all the tasks in the hyper period. The next section presents the adopted solution to remain this problem.

B. Scheduling algorithm choice

Many scheduling algorithms exist such as fixed priority, EDF. We discuss hereafter the use of the fixed priority algorithm. Some problems can occur when the task cannot finish its execution during its period. The graph of the figure 2 illustrates such a case. We consider a system which executes three tasks T1 (period=10, priority=1, Texe=3), T2 (period=10, priority=2, Texe= 3) and T3 (period=15, priority=1, Texe=6).

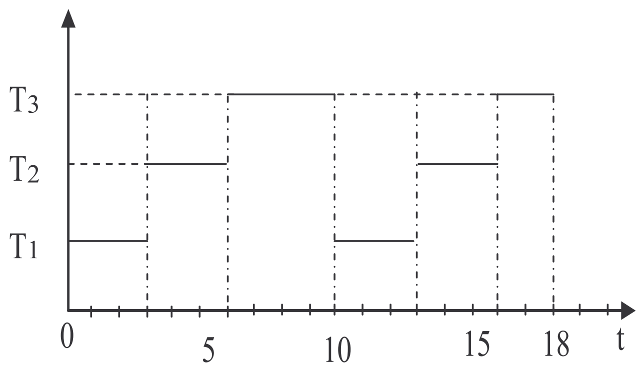

Figure 2. execution graph on MicroC_OS-II

We remark that task T3 exceeded its period to finish its execution (t=18) and this is due to the fact that task T1 and T2 have higher priorities which prevent their interruption. Thus we were brought to seek an adequate scheduling that guarantee that each task is running by all its periods once. Using the EDF algorithm (Earliest Deadline First) based on the dynamic priorities assignment; previously described problem can be avoided. Let us see again the same example where we consider the value of each task deadline equal to its period. The execution graph of the different tasks on an operating system which is based on an EDF scheduling is illustrated by the figure 3.

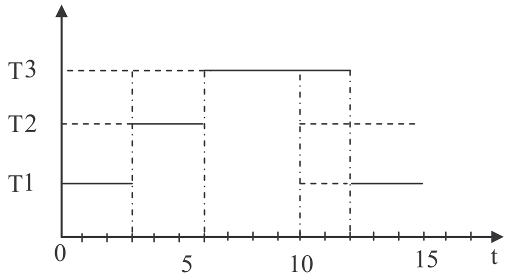

Figure 3. Execution graph using EDF algorithm

We notice that all the tasks finish their execution before the end of their periods. From that, the EDF scheduling meets our needs.

C. *Texe/application parameters models*

Local adaptation is based on models of Texe=f(application parameters) to choose the appropriate applications parameters. We describe in this paragraph their installation. In order to not exceed the deadline, we adopt the worst case execution time (WCET) of the application. Many researches have dealt with this problem [27, 28, 29]. In our framework we adopted a static and dynamic approach to calculate the WCET. The first step consists in analyzing manually the code to retrieve the longest execution path of the application. In the second step we execute these/this critical paths on the adopted experimental platform for several times. The value of WCET is the greatest found value. We repeat those experiments for a set of application parameters values to obtain the Texe=f(application parameters).

V. EXPERIMENTATION

The adopted platform is Excalibur from Altera. It is based on a STRATIX FPGA development board based. We use as RTOS the MicroC_OS-II. Its uses a fixed priority algorithm and not supports the EDF algorithm. We describe hereafter how we implement the EDF.

A. *Establishment of EDF scheduling policy*

EDF is a scheduling policy belonging to the class of dynamic priority algorithms where a task priority changes during its execution. The highest priority task selected by EDF is the one with the earliest deadline.

Our EDF scheduling is hooked into the µC/OSII internal functions so that we are able to switch between it and the initial one (fixed priority).

Since µC/OSII supports only a-periodic tasks while EDF needs periodic ones, we started by enabling task periodicity. We added a data structure in the user definable task control block (TCB) extension, where we defined the task period and deadline with a delay counter for each of them. Figure 4 shows the Task Control Bloc with the extended data structure (Task User Data) used for the EDF scheduling. TCBs are placed in a linked list pointed by the pointer OSTCBList.

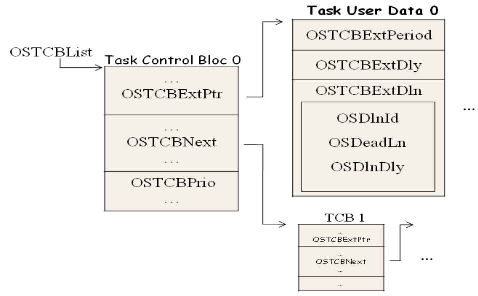

Figure 4. The TCB with the EDF extended data structure

The update of period and deadline delays, and the control of periodic tasks activation have been charged to an internal function called OSTimeTick(). This function is itself triggered periodically by the system ticker ISR. Whenever a scheduling event occurs (ticker ISR, task finishes or is delayed, new task released, etc.) the TCB list will be searched for the task closest to its deadline. This task will then be scheduled for execution next.

B. *Results*

The adopted application is 3D image synthesis. Application parameter are triangle number (triangle_nb) of the 3D generated objet and the shading method (flat and Gouraud). Figure 4 and 5 show time execution model based on measures for the 3D application using both flat and Gouraud. HW implementation corresponds to a one HW accelerator.

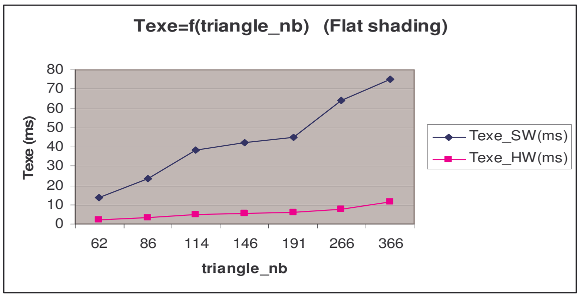

Figure 5. Texe/triangle_nb variation for flat shading

$$Texe\_PlatSW = 18{,}44*10^{-2}*NbPoly + 7{,}71 \quad E\ 3$$

$$Texe\_PlatHW = 2{,}83*10^{-2}*NbPoly + 1{,}10 \quad E\ 4$$

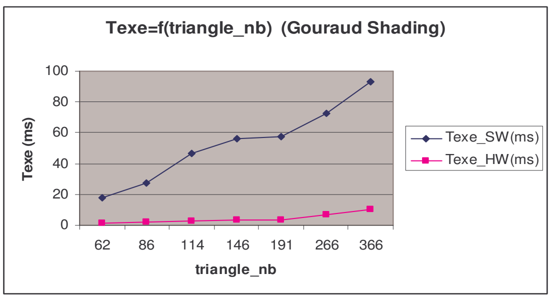

Figure 6. Texe/triangle_nb variation for Gouraud shading

$$Texe\_Gouraud\_SW = 0.205 * NbPoly + 18,34 \quad E5$$

$$Texe\_Gouraud\_HW = 4,27*10^{-2} * NbPoly + 1,23 \quad E6$$

In case where one task exceeds its deadline, the local manager uses E1, E2, E3 or E4 model to determine the application parameters. This choice is based on the actual hardware configuration.

VI. CONCLUSION

Due to their mobility, embedded multimedia systems must be adaptive to ensure maximum QoS under energy and computation constraints. We presented a cross layer adaptation framework. The proposed approach must present a minimal overhead which doesn't degrade the performances of the system. For that, our approach is based on a global manager and a local manager.

This paper focuses on the local manager which intervenes in the OS and application level. This manager satisfies real time constraints by local modification of application parameters. This approach uses a set of constructed models to determine the application parameters from the execution time. We applied our local manager on a reconfigurable platform based on NIOS processor and μC/OS-II RTOS.

References

[1] Wanghong Yuana, Klara Nahrstedta, Sarita V. Advea, Douglas L. Jonesb, Robin H. Kravets "Design and Evaluation of a Cross-Layer Adaptation Framework for Mobile Multimedia Systems" Appears in SPIE/ACM Multimedia Computing and Networking Conference (MMCN), 2003

[2] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," in Proc. of 18th Symposium on Operating Systems Principles, Banff, Canada, Oct. 2001.

[3] R. Melhem, N. AbouGhazaleh, H. Aydin, and D. Mosse, "Power management points in power-aware real-time systems," in Power Aware Computing, R. Graybill and R. Melhem, eds., Plenum/Kluwer Publisher, 2002.

[4] M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for reduced CPU energy," in Proc. of USENIX Symposium on Operating Systems Design and Implementation, 13-23, Nov. 1994.

[5] A. Bavier and L. Peterson, "The power of virtual time for multimedia scheduling," in Proc. of 10th International Workshop for Network and Operating System Support for Digital Audio and Video (NOSSDAV), June 2000.

[6] H. H. Chu and K. Nahrstedt, "CPU service classes for multimedia applications," in Proc. of IEEE Int. Conf. On Multimedia Computing and Systems (ICMCS'99), Florence, Italy, pp. 296–301, June 1999.

[7] S. Banachowski and S. Brandt, "The BEST scheduler for integrated processing of best-effort and soft real-time processes," in Proc. of SPIE Multimedia Computing and Networking Conference, San Jose, CA, Jan. 2002.

[8] M. Mesarina and Y. Turner, "Reduced energy decoding of MPEG streams," in Proc. of SPIE Multimedia Computing and Networking Conference, San Jose, CA, Jan. 2002.

[9] A. Vahdat, A. Lebeck, and C. Ellis, "Every joule is precious: A case for revisiting operating system design for energy efficiency," in Proc. of 9th ACM SIGOPS European Workshop, Kolding, Denmark, Sept. 2000.

[10] K. Loukil, N, Ben Amor, M. Hentati, M. Abid « HW/SW partitioning approach on reconfigurable multimedia system on chip» The 7th ACS/IEEE International Conference on Computer Systems and Applications, Rabat, Morocco. May 10-13, 2009

[11] J. Flinn, E. de Lara, M. Satyanarayanan, D. Wallach, and W. Zwaenepoel, "Reducing the energy usage of office applications," in Proc. of Middleware 2001, Heidelberg, Germany, Nov. 2001.

[12] Pham Ngoc, G. Lafruit, J-Y. Mignolet , G. Deconinck, and R. Lauwereins "QOS aware HW/SW partitioning on run-time reconfigurable multimedia platforms" Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms, ERSA'04, June 21-24, 2004, Las Vegas, Nevada, USA. CSREA Press 2004, ISBN 1-932415-42-4

[13] W. Van Raemdonck, G. Lafruit, E.F.M. Steffens, C.M. Otero Pérez, R.J. Bril "Scalable graphics processing in consumer terminals" Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference

[14] N. Pham Ngoc, W. van Raemdonck, G. Lafruit, G. Deconinck, and R. Lauwereins "A QoS framework for interactive 3D applications" Proceedings of the ninth international conference on 3D Web technology 2004

[15] N. Pham Ngoc, G. Lafrui, G. Deconinck, and R. Lauwereins "Terminal QOS management on run-time reconfigurable platforms" Third PA3CT-symposium 22-23 September 2003

[16] Wanghong Yuan, Klara Nahrstedt "A Middleware Framework Coordinating Processor/Power Resource Management for Multimedia Applications" Proc of Globecom 2001

[17] M. Corner, B. Noble, and K. Wasserman, "Fugue: time scales of adaptation in mobile video," in Proc. of SPIE Multimedia Computing and Networking Conference, San Jose, CA, Jan. 2001.

[18] J. Flinn and M. Satyanarayanan, "PowerScope: A tool for profiling the energy usage of mobile applications," in Proc. of 2nd IEEE Workshop on Mobile Computing Systems and Applications, Feb. 1999.

[19] S. Brandt and G. J. Nutt, "Flexible soft real-time processing in middleware," Real-Time processing in middleware," Real-Time Systems 22(1-2), 2002.

[20] V. Bharghavan, K. Lee, S. Lu, S. Ha, J. Li, and D. Dwyer, "The TIMELY adaptive resource management architecture," IEEE Personal Communications Magazine, 5(4) , Aug. 1998.

[21] K. Loukil, N. Ben Amor, M. Abid "Self adaptive reconfigurable system based on middleware cross layer adaptation model», (IEEE SSD 2009) Djerba, Tunisia 2009.

[22] B. Dave, G. Lakshminarayana, N. Jha, "*COSYN : Hardware-Software CoSynthesis of Embedded Systems,* in Proc. ACM/IEEE Design Automation Conference, Anaheim, CA, June 1997.

[23] W. Fornaciari, P. Gubian, D. Sciuto, C. Silvano, "*Power Estimation of Embedded Systems : A Hardware/Software Codesign Approach*" IEEE Trans. VLSI Systems, June 1998.

[24] W. Fornaciari, D. Sciuto, C. Silvano "Power Estimation for Architectural Exploration of HW/SW Communication on System-Level Buses," In Proc. CODES'99 7th International Workshop on Hardware/Software Co-Design, Roma, Italy, May 1999.

[25] Y. Le Moullec, D. Heller, J.-P. Diguet, J.-L. Philippe, "*Estimation du parall´elisme au niveau syst`eme pour l'exploration de l'espace de conception de syst`emes enfouis*", Technique et Science Informatiques (RSTI-TSI), Lavoisier Hermes-Science publications.

[26] Turley J., "Embedded Processors of Tomorrow ", *Embedded Systems Programming,* novembre 2003.

[27] F. Stappert and P. Altenbernd. "Complete worst-case execution time analysis of straight-line hard real-time programs". Journal of Systems Architecture, 46(4):339{355, 2000.

[28] P. Puschner and G. Bernat. Wcet analysis of reusable portable code. In Proc. Of the 13th Euromicro Conference on Real-Time Systems, pages 45{52, Delft, The Netherlands, June 2001.

[29] P. Puschner and A. V. Schedl. "Computing maximum task execution times a graph based approach". In Proc. of IEEE Real-Time Systems Symposium. Kluwer Academic Publishers, 1997.