Aalborg Universitet

Adaptive Wireless Systems Optimization based on Follow-up Modeling

Le Moullec, Yannick; Ben Amor, Nader; Diguet, Jean-Philippe; Koch, Peter

Published in:
GSPx 2004

Publication date:
2004

Document Version
Peer reviewed version

Citation for published version (APA):
Le Moullec, Y., Ben Amor, N., Diguet, J-P., & Koch, P. (2004). Adaptive Wireless Systems Optimization based on Follow-up Modeling. In GSPx 2004 (Vol. CDROM). GSPx.

Adaptive Wireless Systems Optimization Based on Follow-up Modeling

Yannick Le Moullec
CISS, Aalborg University
Fr. Bajers Vej 7B
9220 Aalborg Ø, Denmark
+45.96.35.98.20
moullec@cs.auc.dk

Nader Ben Amor
LESTER, Université de Bretagne Sud
BP 92116
56321 Lorient Cedex, France
+33.2.97.87.45.28
benamor@iuplo.univ-ubs.fr

Jean-Philippe Diguet
LESTER, Université de Bretagne Sud
BP92116
56321 Lorient Cedex, France
+33.2.97.45.41
diguet@iuplo.univ-ubs.fr

Peter Koch
CISS, Aalborg University
Fr. Bajers Vej 7B
9220 Aalborg Ø, Denmark
+45.96.35.86.96
pk@kom.auc.dk

Abstract. A framework for the modeling and the design of adaptive wireless systems such as software radio and multimedia applications is presented. These applications use adaptive algorithms to minimize the required processing power and to adapt to the system environment. We propose a framework based upon a model that can predict the behavior of a fluctuant system in real-time and that can be used to dynamically perform the application parameterization in order to optimize the trade-off between power consumption, QoS and performances. For this purpose we use the theory of follow-up systems for discrete processes.

1. Introduction

Designing wireless personal communication systems is a difficult task. There is a clear gap between the required performances for the applications, the evolution of VLSI technology and economically viable architectural solutions. Future telecommunication applications such as 4G imply computational power that does not compare favorably with the energy autonomy required. Theoretically, a dedicated, tailored architecture would therefore be a must. However, the increasing circuit costs implies a maximum re-use of hardware and software IPs. Efficiency requirements are often in contradiction with economic constraints which push on flexibility. Moreover, the technological evolution intensifies this tendency, resulting in circuits dominated by interconnections required for flexibility in terms of hardware (e.g., FPGAs) and software (e.g., processor control unit). In addition, power consumption is a critical issue: on the one hand the reduction of transistor sizes enables the reduction of the supply voltage and therefore of the dynamic power consumption (proportional to the squared voltage). On the other hand, this size reduction increases leakage currents and therefore the static consumption (which used to be neglected). Finally, new telecommunication applications (UMTS, 4G) are targeting large amount of data transfers requiring efficient memory resources and coding/decoding architectures, which increases the complexity of the systems.

1.1 Existing works

In the literature several solutions have been proposed to tackle the flexibility versus performance problem. Solutions found at the system level have been proven to be quite efficient since they consider a global approach of the problem while exploiting the configurability of the target architecture. Two main approaches have been recently explored. The first one is the dynamic scheduling of tasks under power consumption and real-time constraints, usually by means of voltage/frequency control [1]. The main idea found in the second one is to shut down the power supply of unused resources by estimating sufficient inactivity periods while respecting the time constraints [2]. These techniques are extended in [3] where a mode-based approach copes with both real time and power constraints. However these approaches are not the ultimate solution [4]. Recently QoS has been considered conjointly with power savings; for example in [5] voltage management is combined with a QoS–based dynamic task scheduler and [6] presents an interesting study about the aggregation of various known techniques targeting several design levels (architecture, Middleware, API) of an end-to-end media delivery QoS/power optimization.

Finally [7] presents a relevant QoS management based on control theory, however the method only addresses the QoS issue via real time scheduling and priority adaptation.

Existing works do not address the issue of algorithmic choices and propose a restricted online power model. Our approach can be seen as a combination of the global view proposed in [6], the feedback priority management detailed in [7] and an online power estimator used to dynamically manage the algorithmic versions of an application.

1.2 Proposition

In this paper we introduce a framework based on a model that can predict the behavior of a fluctuant system in real-time and can be used to dynamically perform the application parameterization in order to optimize the trade-off between power consumption, QoS and performances. The objective of the system is to update dynamically the parameters for the algorithmic configuration while following-up the effects of i) the requirements of the user, ii) the environment and iii) the algorithmic configuration choices. Each application (i.e., threads, processes, tasks…) is identified by an interface specifying the user requirements in terms of energy (battery life-time…) and QoS (frame-rate, picture quality…). Relevant data is then monitored on the system and fed back to the follow-up task. Moreover the regulation process also uses architectural parameters such as the control of the frequency/voltage supply, turning on/off of the peripherals… In addition, some data is collected on the architecture and fed back to the follow-up system: utilization of input/output buffers (i.e., data-rates), energy level, …

As compared to existing works the following elements are considered simultaneously:

- User/System constraints: battery life-time, peak power, tasks QoS…,
- Operating environment: network type, location…,
- Context and energy aware regulation by means of data-metrics and HW/SW reconfigurations (algorithmic and architectural reconfigurations).

Since wireless applications are inherently multi-threaded (user and data communication control, data adaptation, packet processing, security…), a new class of multi-threaded processors for network and wireless applications has been recently introduced [8][9]; in this work we use one of these processors as part of a wireless system case study.

The remainder of the paper is organized as follows: in section 2 details about the proposed model are given. Section 3 presents experimental results illustrating the potential of our approach. Finally section 4 presents our conclusions and perspectives.

2. Follow-up modeling

2.1 Principle

The main idea of the proposed method is to maintain a systematic arranging of algorithmic configurations enabling the selection, for a given average power, of the most suited solution while taking into account the task priorities and the user’s requirements in terms of energy (battery life-time…) and QoS (frame-rate, picture quality…).

The first step is the definition of the interface between the application and the follow-up system. For that purpose it is necessary to identify the input/output parameters which enable i) the application configuration and ii) the measure of the obtained QoS. The second step consists in devising a state model used to predict the evolution of the parameters influencing the system (data-rate, power consumption, QoS of channel decoding…). The resulting state matrix accounts for interactions between parameters and between tasks. Finally, the third step consists in devising a follow-up system to track as precisely as possible the user's requirements in terms of e.g., battery lifetime and QoS. This follow-up system is implemented as a task and is responsible for the algorithmic configuration of the other tasks running on the system. The core feature of the follow-up mechanism is a graph used to order the algorithmic configurations according to the priority and power consumption of the tasks.

There is a trade-off between a fixed and pre-defined organization of the configurations and the handling of hazards. Indeed a certain number of configurations are foreseen according to i) the variation parameters of each task, ii) the value of critical data, and iii) the number of possible algorithmic versions for a given task. However, the (possibly) random nature of the targeted applications would require to foreseen an infinite number of cases to take into account all the possible hazards. As it is almost impossible (and time consuming) to model everything we propose to limit, on the one hand, the number of possible cases and on the other hand to handle data versatility by means of update of the systematic arranging of the configurations. The update process requires access to the relevant data (such as the energy level of the battery) on the system.

2.2 Components of the follow-up model

The proposed model is made of the following components (note that we do not always use all of them, depending on the system complexity):

- TAC[k]: Task Algorithmic Configuration. Array indicating the current algorithmic configurations of the system tasks at time k;

- DM[k]: Data Metrics. Array indicating the value, at time k, of metrics characterizing meaningful data for each task. This point is a non-trivial issue since it requires the designer to analyze his system and select meaningful data-metrics which enables the characterization of the configurations. For this purpose the designer may use DesignTrotter [10] which is currently being updated to perform such a characterization;

- CA[.]: Configuration Associations. Matrix representing allowed/forbidden associations between configurations according to DM values (e.g., no motion correction in MPEG is picture type is I);

- UC[.]: User/System Constraints. Matrix storing the user constraints (e.g., desired battery life-time, maximum instantaneous power…) and the associated tolerances (expressed in percentage);

- TP[]: Task Priorities. Array indicating the task priorities;

- P^[k+1]: Estimation at time k of the average power at time k+1;

- ASG: Algorithmic Selection Graph. Graph-based representation of the systematic arranging of the configurations. This representation is used to store, update and extract configurations for a given power value. A node represents a specific configuration for a given task associated to compatible DMs. The graph is ordered using a left-edge algorithm such as the left branch corresponds to the minimum consumption and the right branch to the maximum consumption. An example is depicted in Fig.1.

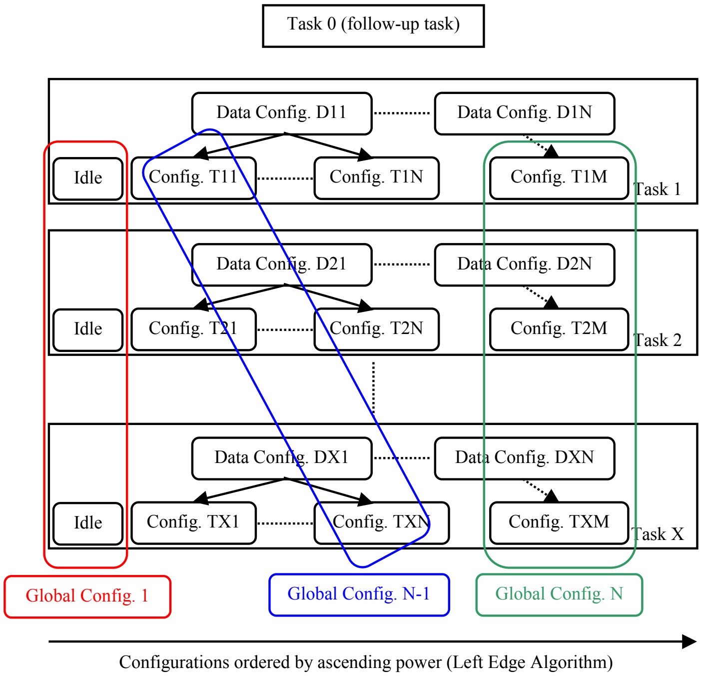

Fig.1. Graph-based representation of the systematic arranging of the configurations.

2.3 Follow-up algorithm

The follow-up mechanism (an example is depicted in Fig.2) is implemented as a task on the system. This task is executed with a period T. The period T depends on a trade-off between the accuracy of the follow-up and the overhead (in terms of time, power…) due to the execution of the follow-up task. The algorithm is as follows:

1. read (for example via a gas-gauge circuit) or estimate the battery level E[k], read data metrics DM,
2. update the content of current active node in the algorithmic selection graph (ASG),
3. re-position the node if necessary (i.e., if the current node power is greater than those of its left neighbors),
4. compute P'[k]. If P'[k] is not equal to zero then continue,
5. compute P^[k+1],
6. extract the configuration complying with P^[k+1].

2.4 Estimating P[k+1]

Knowing E[n] at time k for n = k-1, k-2, …k-M it is possible to i) compute P[k], i.e., the average power consumed in the interval [k-1 ; k], ii) compute P'[k] the discrete derivative of P[k] in the same interval and iii) to estimate E[k+1]. For that purpose several solutions can be considered:

- Kalman filter if E[k] is assumed to be a random noisy signal,
- linear extrapolation: in a regular gas-gauge circuit the energy signal is filtered, in that case a linear extrapolation of the evolution of P can be approximated such as *P[k+1] =Ak+B with A = P'[k] and B=P[k].*

We assume that the evolution of P is locally linear, from where:

$$(1)\quad \hat{E}[k+1]-E[k]=\int_{kT}^{(k+1)T}P(t)dt=\int_{kT}^{(k+1)T}(at+b)dt$$

$$=\frac{1}{2}P'[k]T^2+P[k]T$$

$$where\quad a=P'[k]\quad and\quad b=P[k]$$

$$(2)\quad \hat{P}[k+1]=P'[k]T+P[k]$$

$$(3)\quad P\max[k+1]=\frac{E[k]-E_0}{D_{\min}-kT}$$

3. Case study

We use the system described in Fig.3 as a case study. The system is composed of a network processor (XINC XC1100) and a media processor (LEON core + HW accelerators). The XINC processor features hardware-multithreading capabilities making it quite efficient in terms of equivalent MIPS.
The classification of the power consumption of the algorithmic configurations is achieved by means of a functional power model of the XINC processor [11]. The other part of the system, the media processor, is designed using the methodology introduced in [12].

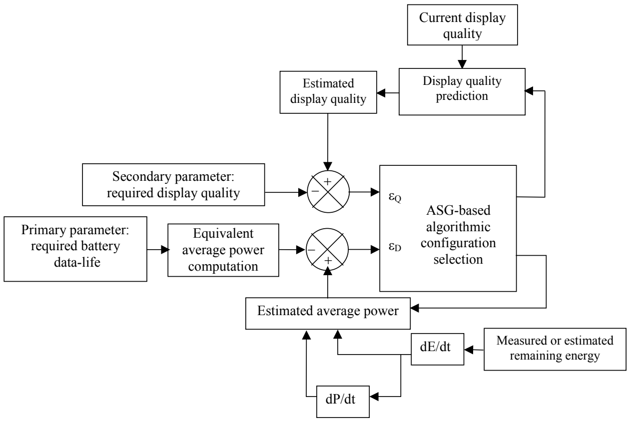

Fig.2. Follow-up model considering two parameters: battery data-life and quality of display. Here the algorithmic configurations are selected according to ε_Q and ε_D which represent the difference between the required and the estimated quality and average power respectively.

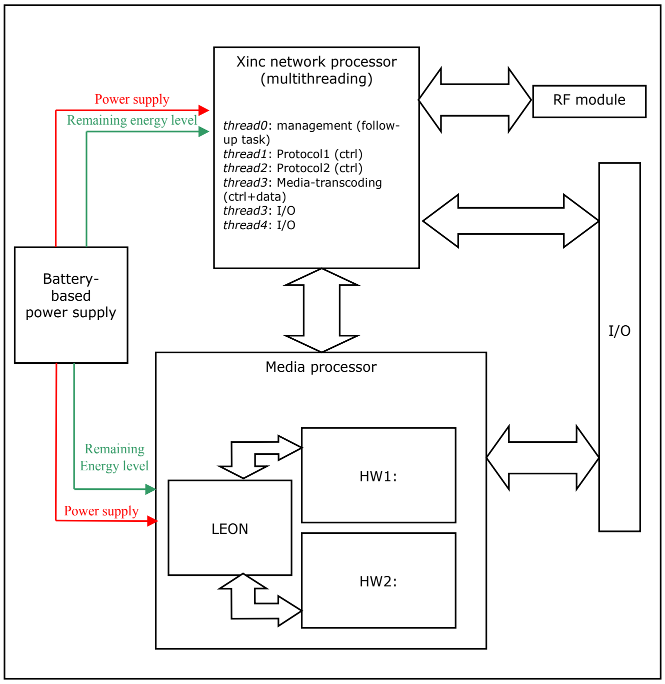

Fig.3. Case study SoC. The XINC is a multi-threaded processor where Thread0 implements the follow-up task described in Fig.2. The other threads implement the application itself (here network protocols and media-transcoding).

NpBench [13], a set of applications from the domain of network processing is used to perform the experiments. This paper reports preliminary results based on experiments conducted with a modified version of the media-transcode module of NpBench.

3.1 Description of the Media-transcode module

The objective of the media-transcode module is to adapt the content of web pages to improve their delivery to a wide range of clients, including portable wireless devices. The module is made of two main steps. The first one is control-plane oriented and is responsible for taking transcoding decisions by taking parameters such as image size, type and purpose, available bandwidth and client display capabilities into account. The second step is data-plane oriented and performs the actual transformations of the data (image re-sizing, color conversion…).

3.2 Experiments

Both the transcoding level and the input data are adjustable, therefore it is possible to generate several algorithmic configurations for the media-transcoding module and thus to explore the QoS/performance/energy trade-off.
The transcoding level is adjustable between 0 (no-transcoding) and 5 (maximum transcoding). When the transcoding level is set to 0 the content is most likely to be un-viewable on a portable device with limited display capabilities. On the other hand, setting the level to 5 will increase the quality at the cost of computation time and energy consumption. The power required by some of the algorithmic configurations on the xinc processor is given in Table.1.

T1 (Media-transcoding) power (mW)					
ver.a:	ver.b:	ver.c:	ver.d:	ver.e:	Ver.f:
284,022	284,200	285,147	286,065	286,443	286,443

Table.1. Power consumption for some algorithmic configurations implemented on the Xinc processor. T1 configurations are related to the transcoding level.

Two scenarios have been used for the experimentation. In the first one the user decides to maximize the readability of the content, whereas in the second one he decides to maximize the battery data-life (i.e., to reduce the energy consumption). The effects of the dynamic reconfiguration performed by the proposed follow-up system are summarized in Table.2. In the first scenario the main objective is to adapt the content to the display capabilities of the device. When enabling the follow-up system, the number of data processed increases by 10% while the consumed energy only increases by 4,66%.

In the second scenario the main objective is to minimize the energy consumption. By using the proposed methodology, the energy consumption can be reduced by 15,8% while the number of data processed only decreases by 8,2%.

At last we can mention that the overhead due to the follow-up task is quite minimal: from a timing point of view it is almost negligible since it is implemented as a thread on the time-slice multithreaded Xinc processor. From a power point of view the overhead is about +0,72%.

	No follow-up	**Follow-up**
Max QoS	1242,02 µJ	1300 µJ
Max life-time	55202,08 µJ	46921,768 µJ

Table.2. Comparison of the energy consumption for two scenarios (max QoS and max battery life-time) with and without the follow-up technique.

4. Conclusions and perspectives

A framework for the optimization of adaptive wireless systems has been presented. In order to optimize the trade-off between power consumption, QoS and performances the proposed framework relies a model that predicts the behavior of a fluctuant system in real-time and dynamically performs the parameterization of the application. In this work the user constraints (battery life-time, tasks QoS), the operating environment (network type, location) are simultaneously considered. This context and energy aware regulation is performed by means of data-metrics and HW/SW reconfigurations (algorithmic and architectural reconfigurations).

A study case has been used to evaluate the proposed methodology and experimental results illustrate the interest of this work. Our perspectives include the refinement of the model for OS aspects and further experimental work, in particular on the OMAP platform from Texas Instrument.

5. References

[1] K. Choi, K. Dantu, W.C. Cheng and M. Pedram, “Frame-Based Dynamic Voltage and Frequency scaling for a MPEG Decoder”, ICCAD, 2002

[2] C.H. Hwang and A. Wu, “A Predictive System Shutdown Method for Energy Saving of Event-driven Computation, DAC 1997

[3] P.H. Chou and N. Bagherzadeh, “Mode selection and mode-dependency modeling for power-Aware Embedded Systems”, ASP-DAC, 2002

[4] B. Zhai, D. Blaauw, D. Sylvester**,** K. Flautner, “Theoretical and Practical Limits of Dynamic Voltage Scaling”, DAC, 2004

[5] J.L. Wong, G. Qu. M. Potkonjak, “An On-line Approach for Power Minimization in QoS Sensitive Systems”, ASP-DAC, 2003

[6] H.V. Antwerpen, N. Dutt, R. Gupta, S. Mohapatra, C. Pereira, N. Venkatasubramanian, R.von Vignau, “Energy-Aware System Design for Wireless Multimedia”, DATE, 2004

[7] J. A. Stankovic, C. Lu, S. H. Son and G. Tao, “The Case for Feedback Control Real-Time Scheduling”, 11th EuroMicro Conference on Real-Time Systems, 1999

[8] XINC XC100, www.elevenengineering.com

[9] UBICOM IP3023, www.ubicom.com

[10] Y. Le Moullec, N. Ben Amor, J-Ph. Diguet, J-L. Philippe and M. Abid, "Multi-granularity Metrics For The Era Of Strongly Personalized SOCs", DATE 2003

[11] E. Baud and C. Leroux, “FLPA Model of the XINC Processor”, internal report, Aalborg University, Denmark, 2004

[12] N. Ben Amor, Y. Le Moullec, I. Maalej, J-Ph. Diguet, M. Abid and J-L. Philippe,” Design of multimedia processor based on metrics computation”, SSD 2003

[13] B. K. Lee and L. K. John, “NpBench: A Benchmark Suite for Control plane and Data plane Applications for Network Processors”, ICCD 2003