Distributed Real-Time Systems with Minimal Energy Consumption: Analysis and Synthesis

(ARTES project 0005-5)
Final report

Krzysztof Kuchcinski, Flavius Gruian
Dept. of Computer Science, Lund Institute of Technology
Box 118, S-221 00 Lund

March 19, 2003

1 Project

The goal of this project was to develop methods for designing real-time systems with low energy and low power consumption. The design methodology is focused from the start, at system level, on energy and power issues, while fulfilling the timing requirements.

1.1 Members

Flavius Gruian starting 2000-08-01 (PhD student)
Krzysztof Kuchcinski (Advisor)

1.2 Comparison between plan and results

Although the project initially addressed static scheduling in multi-processor systems, the emphazis switched to run-time scheduling for single processor systems, mainly because of the available hardware platform used in the experiments. Our initial industry partner, AXIS, turned out to be only long term interested in energy consumption reduction, focusing today on stationary applications. Fortunately we did get support from *Intel*, that was interested in our experiments and donated an evaluation board for a variable speed processor with an XScale architecture. We were therefore able to conduct measurements on a realistic system, validating our scheduling approach.

Briefly, the project focused speed scheduling for dynamic supply voltage processors (such as *Transmeta* Crusoe and *Intel* XScale). Basic real-time scheduling strategies, such as rate-monotonic and earliest deadline first scheduling, were extended with speed management techniques essential to energy consumption reduction. Furthermore, a few novel scheduling techniques were developed both

at task level and at task-set level, designed for tasks with variable execution pattern — a more realistic assumption in the context of real-time systems.

2 Achievements

We regard the achievement of our project to be in two directions, of equal importance. First, the development of new, improved scheduling techniques for low energy consumption and second, the validation of these techniques by implementation and measurements on an actual hardware platform. In the area of speed scheduling for low-energy it is seldom that measurements on real platforms are presented, most of the experiments being carried out using simulators — providing only partial validation of the techniques under scrutiny. Using the evaluation board provided by *Intel* we were able to fully assess the effectivenes of our strategies. More concretely the speed scheduling methods developed in this project are:

- **stochastic scheduling**: a task level speed scheduling technique for tasks with variable execution pattern, that can be implemented exclusively at the RTOS level without prior knowledge about the task internals. By contrast most of the other existing task level speed scheduling methods require compiler and profiler support (XScale validated).
- **maximum required speed**: offline speed selection techniques for both rate monotonic (RM) and earliest deadline first (EDF) strategies, able to achieve full processor utilization and minimal energy consumption for tasks with fixed execution time (XScale validated).
- **RMS with slack distribution**: a speed selection enhanced rate-monotonic scheduling, based on a slack-stealing like slack redistribution strategy. The method can handle both tasks with fixed and variable execution pattern, and can be easily combined with task level strategies.
- **uncertainty-based scheduling**: a novel ordering strategy for task sets, improving the efficiency of the run-time speed manager. The method is based on the statiscs about the task sets gathered both offline and run-time (XScale validated).

All these techniques are designed for hard real-time systems. For the RM and EDF enhanced methods, the response times are kept. Since variable speed processors are already making their way into the consumer market (laptops, PDAs), these strategies are able to reduce the system energy consumption (thus cost) in all kinds of applications.

3 Publications

- Gruian, F.; Kuchcinski, K., **Uncertainty-Based Scheduling: Energy-Efficient Ordering for Tasks with Variable Execution Time**, sub-

mitted to the International Symposium on Low Power Electronics and Design 2003.

- Gruian, F. **Energy-Centric Scheduling for Real-Time Systems**, Doctoral dissertation 15, ISBN 91-628-5494-1, ISSN 1404-1219, 2002, Lund.
- Gruian, F. **Hard Real-Time Scheduling for Low-Energy Using Stochastic Data and DVS Processors**, International Symposium on Low Power Electronics and Design 2001, Huntington Beach (CA), US, August 6-7, 2001.
- Gruian, F. **On Energy Reduction in Hard Real-Time Systems Containing Tasks with Stochastic Execution Times**, IEEE Workshop on Power Management for Real-Time and Embedded Systems, Taipei, Taiwan, May 29, 2001, pp.11-16.
- Gruian, F.; Kuchcinski, K., **LEneS: Task-Scheduling for Low-Energy Systems Using Variable Voltage Processors**, Asia South Pacific - Design Automation Conference 2001, January 30 - February 2, Yokohama, Japan.
- Gruian, F. **System-Level Design Methods for Low-Energy Architectures Containing Variable Voltage Processors**, Power-Aware Computing Systems 2000 Workshop, November 12, Cambridge (MA), US. Also available in LNCS2008, p.1.

4 Industry statement

The fact that *Intel* provided us with a free board for our experiments shows the industrial interest in our work. We carried out discussions with Bill E. Brown and Atila Alvandpour from *Intel*, regarding hardware aspects specific to variable speed processors. For example we arrived at the conclusion that speed switching time overhead, significant from the real-time perspective, may be greatly reduced if one employs better clock generation architectures (multiple DLLs). On another note, it turned out that measuring real-time on the evaluation board was a hassle, especially during speed switches.

The experimental platform is still used in our laboratory, for porting the ivm java virtual machine, enhanced with a run-time speed manager. Our contact with *Intel* is continuing, and we are hoping to receive a new donation in the form of an improved evaluation board for the XScale architecture.