Exploring Chip-Multiprocessors in Deeply-Embedded Real-Time Computing

Xuan Qi
Advisor: Dakai Zhu
Department of Computer Science
University of Texas at San Antonio
San Antonio, TX, 78249
{xqi, dzhu}@cs.utsa.edu

1 Abstract

As an energy efficient high-performance architecture, *chip multiprocessor (CMP)* can be deployed in deeply-embedded real-time computing. In this essay, I propose the application of CMP in deeply-embedded real-time system. I discuss the issues to be addressed such as effective resource modeling, efficient scheduling algorithms, and energy efficient design. And the system and experiment setup to evaluate our model and algorithms will be proposed.

2 Introduction

As the integration of computing systems with the physical environment becomes seamless, in addition to logical correctness, computing systems will also face physical constraints of time, space and resources (such as energy). For such deeply embedded systems, which generally involve extensive computation and communication, it is a great challenge to deliver the required high performance with limited system resources. As the communication becomes more significant in the widely deployed distributed architectures, another challenge is how to efficiently schedule the required communication and computation in the ever-increasing complex systems. Moreover, for safe-critical deeply-embedded applications (such as automatic medical assistant robots), the ability to tolerate occasionaly occurred faults is foremost important. The conflicting requirements on performance, energy efficiency and reliability demand new design paradigm for such deeply embedded systems.

As one energy efficient high-performance architecture, CMP has been proposed to improve system performance with multiple simpler processing cores on a single chip, where each core may have multiple thread running contexts with *simultaneous multithreading (SMT)* techniques. The central idea for CMP/SMT processors to improve system performance and power efficiency is to exploit both instruction- and thread-level parallelisms in the applications. In addition, the inherent hardware redundancy in CMP processors provides the opportunity to design flexible fault tolerance schemes. Considering its special features on high performance, power efficiency and fault tolerance, CMP has been adopted by several chip manufacturers (e.g., AMD and Intel) as the technology for the next generation processors. With the inherent parallel nature of the deeply embedded applications (such as sensored data processing), it is expected that CMP/SMT processors will be widely depolyed in such systems for high performance, energy efficiency and high reliability.

However, for the tightly-coupled CMP/SMT processors where processing cores may share L2 cache and even functional units, the problem of *nonpredictable performance* for applications executing on CMP/SMT-based systems has been identified. Such performance nonpredictability could lead to serious problems, especially for deeply embedded *real-time* systems, where *timeliness* is an important requirement and the consequence of missing a deadline could be catastrophic (e.g., the humanoid controller). Therefore, special attentions are needed when exploiting the CMP/SMT technology in such timing constrainted systems.

3 Research Problems

Effective Resource Modeling Techniques We believe that effective resource modeling and management techniques, which consider both the special features in CMP/SMT processors and detailed resource requirements of the applications, are the *key* components to efficiently exploit CMP/SMT processors in deeply embedded real-time systems. First, I will investigate various real-time resource modeling and management techniques for CMP/SMT processors. Such techniques will form the basis for designing high performance, energy efficient and reliable CMP-based deeply embedded real-time systems.

For conventional processor based real-time systems, CPU is generally modeled as a single-dimension *time-multiplexed* resource, which is *exclusive* accessed by real-time tasks. The timing constraints are generally guaranteed through *the worst case analysis* and effective *resource reservation* mechanisms. Considering the tightly-coupled resources (e.g., the instruction window, functional units and on-chip caches) that could be utilized/shared simultaneously by different threads/tasks in CMP/SMT processors, the execution time of any thread will depend on the resource requirements and execution behaviours of other concurrent running threads. When threads compete for resources, they may interact poorly. Conversely, for threads that need *complementary* resources (e.g., one integer-intensive thread and one FP-intensive thread), they may run without noticeable interferences with each other and the resource utilization will be maximized. Therefore, I plan to study a *fine-grained multidimensional* resource model, which will consider both *temporal* and *space* (e.g., different functional units and multiple processing cores) dimensions for CMP-based systems.

Efficient Scheduling Algorithms for High Performance Based on the to-be-developed resource and execution model, I will investigate efficient co-scheduling algorithms that consider both resource utilization and timing constraints for various real-time task models. Specifically, relying on the fine-grained multidimensional resource models, I will first extend *symbiosis factor* concept and investigate detailed symbiosis modeling techniques. Then, the performance effects (e.g., *slow-down factors*) for various symbiotic scenarios among real-time tasks will be examined.

In addition to the co-scheduling algorithms that deal with tasks/threads running on the *same* core, at a higher level, I will also look at the problem how to efficiently schedule tasks among different processing cores. Extending the well-known optimal algorithm, the *proportional fair (Pfair)*, for scheduling periodic tasks on multiple processors, the *boundary fair (Bfair)* scheduling algorithm has been studied, which makes scheduling decisions and ensures fairness for tasks *only* at period boundaries. Following this line of research, to further reduce scheduling overhead, I plan to extend Bfair scheduling for other scenarios considering both the communication and computation requirements in the CMP-based deeply embedded real-time systems.

Energy Efficient Design Next step, I plan to investigate energy efficient management schemes in CMP-based deeply embedded real-time system. Developing reasonable and accurate system wide power/energy model is our first goal. Following that, we will further work on static and dynamic power management schemes for CMP-based deeply-embedded real-time system.

If time permitted, flexible fault tolerance schemes will also be studied by considering the inherent hardware redundancy of CMP processors as well as temporal redundancy in real-time applications.

4 System and Experiment Setup

To evaluate the proposed models and algorithms, we will extend the CMP simulator GEMS, which runs on top of of Simics, an execution-driven full-system functional simulation environment. Moreover, the models and algorithms will be validated through physical experiments, in which real applications will be executed on an Intel Core 2 Duo (C2D) based development testbed. Finally, the C2D-based testbed will be incorporated and tested in a mobile robot (as a deeply embedded real-time system) for data collection in a sensor network.