EVALUATION AND MAPPING OF APPLICATIONS ON HETEROGENEOUS MULTIPROCESSOR SYSTEMS

ARYABARTTA SAHU

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY DELHI
JULY 2009

EVALUATION AND MAPPING OF APPLICATIONS ON HETEROGENEOUS MULTIPROCESSOR SYSTEMS

by

ARYABARTTA SAHU

Department of Computer Science and Engineering

Submitted in fulfillment of the requirements of the degree of

Doctor of Philosophy

to the

Indian Institute of Technology Delhi
July 2009

Certificate

This is to certify that the thesis titled **Evaluation and Mapping of Applications on Heterogeneous Multiprocessor Systems** being submitted by **Aryabartta Sahu** for the award of **Doctor of Philosophy** in **Computer Science & Engg.** is a record of bona fide work carried out by him under our guidance and supervision at the **Deptt. of Computer Science & Engg., Indian Institute of Technology Delhi**. The work presented in this thesis has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma.

M. Balakrishnan
Professor
Deptt. of Computer Science & Engg.
Indian Institute of Technology Delhi

Preeti Ranjan Panda
Associate Professor
Deptt. of Computer Science & Engg.
Indian Institute of Technology Delhi

Acknowledgments

I am greatly indebted to my supervisors Prof. M Balakrishnan and Dr. Preeti Ranjan Panda for their valuable technical guidance and moral support. I would like to thank Prof. Anshul Kumar, Dr. Kolin Paul, Prof. Subhashis Benerjee and Prof. G.S. Visweswaran for their valuable feedback, suggestions and help in all respect. I would like to thank Anant Vishnoi, Nagaraju Pothineni, Neeraj Goel and Vikram Goyal for their feedback during our informal discussions and my other colleagues specially V.N. Muralidhara, Uma mudengudi, Smruti Padhy, Ayesha Choudhury, Lava Bhargava, Sonali Chouhan, Kameshwar Rao, G. Krishnaih and B.V.N. Silpa for their cooperation and support. I would also like to thank the staff of Philips, FPGA and Intel laboratories, IIT Delhi for their help. I am very thankful to Naval Research Board, Govt. of India for being enabler of my work.

My family members showed immense patience and provided me great moral support during the course of my work.

Aryabartta Sahu

Abstract

In recent years, embedded system applications mostly pertain to image processing, vision and wireless networking domains. These applications exhibit high degree of parallelism apart from being very compute intensive in nature. The implication is that multiprocessor solutions are likely to be implementation platforms for embedded applications that are software dominated. Multiprocessor solutions imply a huge increase in the design space from which an optimal configuration is to be selected at design time. This is because even for a fixed processor, the number of possible configurations due to the large number of configuration parameters like memory size, cache, interconnection switches, synchronization techniques, etc. contribute to the design space. Efficient performance estimation based exploration is essential to be integrated into the design framework to effectively explore this vast design space.

In this work, we describe a methodology for retargetable (i.e., user specified target processor and architecture) software estimation to enable design space exploration of heterogeneous multiprocessor systems.

The work reported in the thesis is divided into four parts:

- Software estimation of C programs on a single processor,
- Software estimation of multi-threaded programs on multiprocessors,
- Software estimation of parallel applications, expressed in Cilk on multiprocessors, and
- A framework for multiprocessor design space exploration targeted for Cilk programs.

For implementing an embedded application, there are varieties of commercially available processors, each one offering a different design point based on factors such as performance and power consumption. Though this thesis deals with multiprocessor design space exploration, selection of a suitable processor remains a crucial activity to design a multiprocessor.

Retargetable software estimation on uni-processor starting from "C" programs takes two inputs: processor description in HMDES (High-level Machine DESciption) and "C"

description of the task. Our estimator generates a set of outputs associated with the task behavior on the processor including the execution time. Execution time estimation of a task mapped onto the target processor is carried out by correlating task requirements with the processor capabilities. The methodology proposed includes the effect of basic compiler optimizations and integrates light weight memory simulation and instruction mapping to improve the accuracy of estimates. In this process, we have used SUIF (Stanford University Intermediate Format) compiler framework to develop a uni-processor performance estimator. Results of various benchmarks on a range of processors using this technique shows that it is possible to predict the performance with a maximum error of around 14%. We have used this framework and methodology to develop software estimation for multiprocessors.

In the second part of our work, we describe software estimation of multi-threaded programs on a multiprocessor. Our methodology takes three inputs; fork-join representation of application, high level description of the multiprocessor target architecture and binding (showing mapping of application components onto architecture resource elements). Output is the application performance on a target multiprocessor architecture. Our framework assumes all application components like computation tasks as well as communication and synchronization protocols are specified in "C" language. Further, it is assumed that all these tasks and constructs execute on an instruction set processor (in contrast to Application Specific ICs (ASICs)). Uni-processor software estimation has been used to estimate the task execution time for each individual processor. To these estimates, delays due to communication and synchronization protocol have been separately estimated and added. To estimate performance degradation due to shared resources like memory and bus, "appropriate" access traces from multiple sources are generated using the uni-processor framework. For the shared resources, interval analysis technique is employed on this "synthetic" traces to predict delays due to contention. Results using different application programs show that using such an approach, it is possible to predict performance with average errors of around 11%. Further, we conclusively show that performance not only depends on the architecture but also depends heavily on the binding of application components onto multiprocessor resources.

Multiprocessing and use of MPSoC are growing rapidly in the embedded systems space. On the other hand, writing and debugging of parallel multi-threaded code is

not only difficult and cumbersome, but also results in inefficient code. Many parallel programming languages have been developed that assist writing and debugging of multi-threaded code. Cilk is one such language developed at MIT in mid-90s. Cilk uses an optimal run time scheduler to map the application on to the multiprocessor architecture. In this work, we have developed a novel methodology for performance estimation of Cilk programs on multiprocessors that can be used in the early phase of design space exploration. The methodology incorporates retargetable (i.e. user specified target processor and architecture) estimation of Cilk multi-threaded applications on MPSoC. We start with generation of dynamic process spawn graph and timing annotation of that graph. We then perform retargetable estimation of task transfer time, task spawn time and task sync time by instrumenting the source code of Cilk run time scheduler and Pthread library. This enables our estimator to capture the behavior of the run time scheduler of Cilk. For validation purposes, we have ported Cilk generated multi-threaded code to a multiprocessor simulator. Results show that using estimation, it is possible to predict the performance with an average error of 19% but with a speed up of over 300x vis-a-vis simulation.

Finally, we use the above methodology to create a framework to explore the multiprocessor architectural space. This framework has been used to generate optimal architetcure for a number of benchmarks. The benchmarks are specified in Cilk and multiprocessor architectures in HMDES. The framework uses a genetic algorithm in exploring the vast design space and identifying suitable candidates meeting the performance constraints for detailed simulations.

Contents