A GENERALIZED EXECUTION MODEL FOR NESTED DATA-PARALLEL COMPUTING

By
Dean Engelhardt
April 18, 1997

A THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ADELAIDE

Contents

Preface

Of the many attempts to simplify the task of programming today's parallel multiprocessor architectures, the most successful is the paradigm of Data-Parallelism. Much of the appeal of the model lies in its high-level view of the parallel machine coupled with its efficient mapping to a large class of real-world architectures.

While the Data-Parallel model has been very effective at allowing highly parallel specification of operations across rectangular arrays, its applicability to programs which use less regular data structures is very limited. This factor compromises the utility of the model: many important scientific problems based heavily upon irregular data structures (e.g., sparse matrix computations, finite element irregular mesh codes) cannot be efficiently parallelized using existing Data-Parallel techniques.

The purpose of this thesis is to explore extensions to the Data-Parallel model which make it more amenable to these kinds of irregular problems. In particular we consider the paradigm of Nested Data-Parallelism proposed by Blelloch, investigating techniques for efficiently mapping programs with such nested parallelism onto traditional multiprocessor architectures. Through mathematical analysis we derive a novel implementation of such features which makes use of a multi-threaded model of computation upon each processor of a multiprocessor machine. We describe a realization of this execution model which we have constructed for the Thinking Machines CM-5 and detail how we have used this system as the basis for the implementation of a simple language with Nested Data-Parallel features. To demonstrate the validity of our approach we benchmark the performance of several programs compiled using this language system, comparing the figures obtained with those for equivalent programs compiled in the CM Fortran and NESL systems. For the real-world irregular codes we examine, our unoptimized compiler output running on the CM-5 threading system delivers performance that is competitive with both existing systems, surpassing each in certain situations.