PARALLELIZING IRREGULAR AND POINTER-BASED COMPUTATIONS AUTOMATICALLY: RESULTS IN LOGIC AND CONSTRAINT PROGRAMMING*

MANUEL HERMENEGILDO†

(Abstract)

Irregular computations pose some of the most interesting and challenging problems in automatic parallelization. Irregularity appears in certain kinds of numerical problems and is pervasive in symbolic applications. Such computations often use dynamic data structures and also often make heavy use of pointers. This complicates all the steps of a parallelizing compiler, from independence detection to task partitioning and placement. Starting in the mid 80's there has been significant progress in the development of parallelizing compilers for logic programming (and, more recently, constraint programming) resulting in quite capable parallelizers. The typical applications of these paradigms frequently involve irregular computations, and make heavy use of dynamic data structures with pointers, since logical variables represent in practice a well behaved form of pointers. This arguably makes the techniques used in these compilers potentially interesting. In this paper we introduce in a tutorial way some of the problems faced by parallelizing compilers for logic and constraint programs and provide pointers to some of the significant progress made in the area. In particular, this work has resulted in a series of achievements in the areas of inter-procedural pointer aliasing analysis for independence detection, cost models and cost analysis, cactus-stack memory management, techniques for managing speculative and irregular computations through task granularity control and dynamic task allocation (such as work-stealing schedulers), etc.

Keywords: Automatic Parallelization, Irregular Computations, Speculation, Cactus Stack, Work Stealing, Pointer Aliasing Analysis, Task Granularity Control, Global Analysis, Abstract Interpretation.

REFERENCES

[1] J. Chassin and P. Codognet, *Parallel Logic Programming Systems*, Computing Surveys, 26 (1994), pp. 295–336.

[2] G. Gupta, E. Pontelli, K. Ali, M. Carlsson, and M. Hermenegildo, *Parallel Execution of Prolog Programs: a Survey*, ACM Transactions on Programming Languages and Systems, 23 (2001), pp. 472–602.

[3] M. Hermenegildo, *Parallelizing Irregular and Pointer-Based Computations Automatically: Perspectives from Logic and Constraint Programming*, Parallel Computing, 26 (2000), pp. 1685–1708.

*This work was supported in part by MCyT projects EDIPIA and ADELA.

†School of Computer Science, Technical University of Madrid (UPM), Spain, herme@fi.upm.es, http://www.clip.dia.fi.upm.es/~herme