Session 12: Parallel Programming.

Chairman: Peter Milligan

Department of Computer Science, Queen’s University of Belfast, U.K.

The drive to identify solutions to the many problems associated with the development of code for execution on parallel architectures continues to dominate research in this area. Apart from the problems associated with the lack of general purpose development environments there is the simple fact that once a parallel version of a program is produced there is no guarantee that it will have a satisfactory execution profile. Certain areas of programming, e.g. logic programming, pose highly specialized problems and there is a long standing and growing interest in solving these problems in a highly efficient manner.

The papers contained in this section of the workshop programme can be divided into two groups. The first contains a paper which will have a broad range of applicability as it suggests how programs exhibiting fine-grain characteristics can be restructured to produce coarse-grain equivalents with a much improved performance profile. The second group of three papers have logic programming as a central feature and consider various performance and implementation related issues.

The first paper by Justo and Welsh from the Centre for Parallel Computing at the University of Westminster and the Computing Laboratory at the University of Kent addresses the problem of improving the performance of programs running on parallel systems. The authors argue that while parallelism simplifies and clarifies the development of complex systems it does not necessarily produce programs that have fast execution times. They pursue this argument by suggesting that program performance can be improved by merging fine-grain processes into larger coarse-grained objects which are more efficient. They refer to this merging process as serialisation.

Following a standard introduction the paper is organized into three main sections, namely one which introduces what the authors refer to as serialisation principles, one which describes the experiments that have been used to investigate this approach to program development and one which presents the resulting performance figures and an analysis of these results. The authors have been careful to draw somewhat cautious conclusions. They indicate that certain examples show that the serialisation process can result in improvements of more than 600% in speed.

The second paper by Peter Kacsuk from the KFKI Research Institute of the Hungarian Academy of Sciences addresses the topic of wavefront scheduling in LOGFLOW. The LOGFLOW project is one of a number of projects aimed at investigating the implementation of logic programming languages on distributed memory systems. LOGFLOW is concerned with the implementation of CPA Prolog. The author argues that LOGFLOW is unique in that it is based on a data driven model. In addition the model is highly optimized for execution on a neighbourhood oriented processor space. The author indicates that the main contribution of the paper is the description of a dynamic scheduling scheme. This scheme is able to control the distribution of Prolog work in the processor space based on the actual workload on the processors. In this sense it can be viewed as an investigation of dynamic load balancing.

The paper overviews the general research area and the LOGFLOW project. The core of the paper are the two sections in which the scheduling process is examined. In the first of these sections the scheduler, described as a wavefront scheduler, is described and in the second the requirement for and the use of a scheduling threshold register are described.

The penultimate paper by Clematis from the Istituto per la Matematica Applicata - CNR and Gianuzzi from the Dipartimento di Informatica e Scienze dell’Informazione Genova considers the area of distributed reactive programming. The authors introduce two different semantics for distributed reactive programs. They refer to these as horizontal semantics and vertical semantics. The core of the authors work is the investigation of implementation strategies for distributed reactive programs on a general purpose distributed parallel architecture. Furthermore they investigate the different semantic approaches by implementing them using what they refer to...
as the Linda coordination mechanism and they determine the useful of this implementation strategy.

The authors commence their paper with an overview of reactive programming and proceed to demonstrate how a reactive program can be represented as a general control loop. From this point they introduce the topics of horizontal semantics and vertical semantics. This is followed by an overview of Linda and an explanation of the implementation strategy used for the different semantic models. The paper concludes with an evaluation of the semantic models.

The final paper in the session by Ciampolini, Lamma, Mello and Stefanelli from the Dipartimento di Elettronica, Informatica e Sistemistica at the Universita di Bologna is concerned with the distributed implementation of a simple OR-parallel logic language. The authors focus on the problems surrounding distributed unification, e.g. excessive inter-node communications. The authors have implemented their language on a transputer based architecture.

The authors introduce their work by immediately focussing on the problems caused by the use of message passing communications as this has an obviously negative effect when considering unification in a distributed environment. To overcome this problem the authors propose a scheme based on flexible rather than fixed copying. In theory this enables the authors to determine an optimal solution for each program. The paper introduces the distributed execution model adopted by the authors and presents some experimental results derived by isolating what the authors refer to as basic time parameters. The authors have implemented their work on a Meiko Computing Surface and discuss the timing results.

The papers in this section are reflective of the interesting and high quality work being carried out in developing efficient parallel programs. Although the majority of the papers address the area of parallel logic programming the core theme of efficiency will make the session interesting to a wide audience.