Workshop 03: Scheduling and Load Balancing

Susan Flynn Hummel, Graham Riley and Rizos Sakellariou

Co-chairmen

Introduction

Mapping a parallel computation onto a parallel computer system is one of the most important questions for the design of efficient parallel algorithms. In the case of irregular data structures, the problem of initially distributing and then maintaining an even workload as computation progresses, becomes very complex. This workshop will discuss the state-of-the-art in this area.

Besides the discussion of novel techniques for mapping and scheduling irregular dynamic or static computations onto a processor architecture, a number of open problems are of special interest for this workshop: One is the development of partitioning algorithms used in numerical simulation, taking into account not only the cutsize but also the special characteristics of the application and their impact on the partitioning. Another topic of special relevance for this workshop concerns re-partitioning algorithms for irregular and adaptive finite element computations that minimize the movement of vertices in addition to balancing the load and minimising the cut of the resulting new partition. A third topic is the development of dynamic load balancing algorithms that adapt themselves to the special characteristic of the underlying parallel computer, easing the development of parable applications.

Other topics, in this uncomplete list of open problems, relate to application-specific graph partitioning, adaptable load balancing algorithms, scheduling algorithms, novel applications of scheduling and load balancing, load balancing on workstation clusters, parallel graph partitioning algorithms, etc.

The Papers

The papers selected for presentation in the workshop were split into four sessions each of which consists of work which covers a wide spectrum of different topics. We would like to thank sincerely the more than 20 referees that assisted us in the reviewing process.

Rudolf Berrendorf considers the problem of minimising load imbalance along with communication on distributed shared memory systems; a graph-based technique using data gathered from execution times of different program tasks and memory access behaviour is described. An approach for repartitioning and data remapping for numerical computations with highly irregular workloads on distributed memory machines is presented by Leonid Oliker, Rupak Biswas, and

Harold Gabow. Sajal Das and Azzedine Boukerche experiment with a load balancing scheme for the parallelisation of discrete event simulation applications. The problem of job scheduling on a parallel machine is tackled by Christophe Rapine, Isaac Scherson and Denis Trystram, who analyse an on-line time/space scheduling scheme.

The second session starts with a paper by Wolf Zimmermann, Martin Middendorf and Welf Loewe who consider k-linear scheduling algorithms on the LogP machine. Cristina Boeres, Vinod Rebello and David Skillicorn consider also a LogP-type model and investigate the effect of communication overheads. The problem of mesh partitioning and a measure for efficient heuristics is analysed by Ralf Diekmann, Robert Preis, Frank Schlimbach and Chris Walshaw. Konstantinos Antonis, John Garofalakis and Paul Spirakis analyse a strategy for balancing the load on a two-server distributed system.

The third session starts with Salvatore Orlando and Raffaele Perego who present a method for load balancing of statically predicted stencil computations in heterogeneous environments. Fabricio Alves Barbosa da Silva, Luis Miguel Campos and Isaac Scherson consider the problem of parallel job scheduling again. A general architectural framework for building parallel schedulers is described by Gerson Cavalheiro, Yves Denneulin and Jean-Louis Roch. A dynamic loop scheduling algorithm which makes use of information from previous executions of the loop is presented by Mark Bull. A divide-and-conquer strategy motivated by problems arising in finite element simulations is analysed by Stefan Bischof, Ralf Ebner and Thomas Erlebach.

The fourth session starts with a paper by Dingchao Li, Akira Mizuno, Yuji Iwahori and Naohiro Ishii that proposes an algorithm for scheduling a task graph on an heterogeneous parallel architecture. Mario Dantas considers a scheduling mechanism within an MPI implementation. An approach of using alternate schedules for achieving fault tolerance on a network of workstations is described by Dibyendu Das. An upper bound for the load distribution by a randomized algorithm for embedding bounded degree trees in arbitrary networks is computed by Jaafar Gaber and Bernard Toursel.