Workshop 04+08+13

Parallel and Distributed Algorithms

Workshop 04+08+13: Parallel and Distributed Algorithms

Keith Marzullo and Cynthia Phillips

The 16 papers in this workshop represent papers submitted in three areas: theory and models of parallel computation, parallel discrete algorithms, and distributed systems and algorithms. These papers share the common theme of searching for algorithms which run efficiently on real machines, and building theoretical foundations for analysis and comparison of these algorithms. Within this common theme, however, the papers represent the diversity of the field: new models of parallel and distributed computation; better algorithms and analyses within known models where "better" is frequently a simpler algorithm or one which makes fewer assumptions about the underlying system; first-time NC parallelizations; and experimental analyses.

There is still no agreement on a model of parallel (tightly coupled) computation. This decade has seen the introduction of many variants of the historical PRAM models, as well as the currently popular BSP and LogP models and variants thereof. The search continues for a model that is just complicated enough to give proper guidance on algorithmic choice on a wide variety of parallel machines. Two papers in this workshop make strong contributions to this search: the paper by Wachsmann and Wanka, which shows the promise of analysis using multilinear functions, and the paper by Sanders, Vollmar, and Worsch, which argues that the (in)ability of higher-dimensional architectures to dissipate power can fundamentally limit their computational power.

Given the difficulty of choosing a model of parallel computation, increasing emphasis has been placed on implementation both to determine the performance of a particular algorithm on a particular machine and to test the efficacy of particular parallel models. The algorithms community is willing to consider an algorithm with poorer worst-case analysis compared to previous algorithms if it is simpler and has experimental analysis to back up its practicality. There is a trend in all of computer science to be more clever, careful, and rigorous in performing experimental analyses, as indicated by the foundation of the online ACM Journal of Experimental Algorithmics. In this workshop, the paper by Bougé, Gabarró, Messeguer, and Schabanel also demonstrates a new trend to use experimentation as an algorithmic design tool. Their analysis reveals interesting structure of the behavior of their AVL-tree rebalancing algorithm.

The popularity of the BSP and LogP models has led to a rush to adapt, develop, and analyze algorithms for these models. However, this workshop represents algorithmic advances in fundamental problems for many different models. There are new sorting algorithms for the CROW PRAM, the mesh, and as analyzed by multilinear functions, and priority queue algorithms for both BSP and the CREW PRAM. One unique contribution in the sorting paper by Sibeyn is inclusion of second-order terms in the analysis.

The simpler PRAM is still the model of choice when developing first-time highly parallel algorithms. Though in practice one rarely has a number of processors even linear in the size of the problem, inclusion in NC is still a good first indication of the parallelizability of a problem. This workshop has two first-time NC algorithms: Liang's paper on 2-connectivity augmentation in a graph, and Serna and Xhafa's paper on approximate scheduling of unrelated parallel machines. The latter paper shows that it is possible to parallelize a technique recently used in many sequential combinatorial approximation algorithms, namely using a linear-programming relaxation as a starting point and as a lower bound for analysis.

Research in distributed computing has focused on the development of robust algorithms for computing environments where machines or communications can be faulty. The goal is to develop provably correct and robust algorithms (given a failure model) that are simple, that use the weakest possible atomic operations, and that make reasonably weak assumptions about system properties. Papers in this workshop represent a new modeling formalization, improved algorithms for combinatorial problems and tools that serve as building blocks for other algorithms, and continued investigations into primitives needed to perform higher-level service.

Self-stabilizing algorithms are extremely robust because no matter what state they start in, they will eventually converge to a legal state. The first self-stabilizing protocol, by Edsgar Dijkstra, was for token passing, and in this workshop a paper by Petit and Villain gives a new protocol for self-stabilizing token passing that improves the state of the art. A second paper, by Antonoiu and Srimani, gives a self-stabilizing protocol for computing a minimimum spanning tree.

Two papers address the issues of what primitives are needed to perform desired higher-level service. Agrawal, Alonso, El Abbadi, and Stanoi discuss the limits of building transaction-level primitives for replicated databases using reliable broadcast primitives. This is an appealing problem, since reliable broadcast protocols seem very suitable for the problem, but as the authors discuss, their application for real systems is not straightforward. Skubiszewski and Porteix show how to build distributed garbage collectors using properly-timed snapshots of object-oriented databases using cuts that are weaker than consistent cuts.

There is one paper on a distributed system: Kalantery's paper on a significantly more efficient version of the parallel discrete event simulator Time Warp. The Time Warp simulation technique uses a liberal amount of optimistic execution to shorten the running time of a simulation. Kalantery's paper describes how they improved the Time Warp technique via a number of optimizations, some of which apply to traditional systems and some of which depend on virtual channels which preserve message order.