International Journal of Computational Geometry & Applications Vol. 2, No. 2 (1992) 113–116

Guest Editor's Introduction

MICHAEL T. GOODRICH
Department of Computer Science, Johns Hopkins University
Baltimore, MD 21218, USA, e-mail:goodrich@cs.jhu.edu

A fundamental goal of parallel computation research is to design parallel algorithms that use as few processors as needed to solve a problem faster than what is possible on a sequential computer. If one can design an algorithm whose running time $T(n)$ and processor bounds $P(n)$ achieve the relationship $T(n)P(n) = O(S(n))$, where $S(n)$ is the time complexity of the best sequential algorithm, then one says that the algorithm achieves a *linear speedup*. This, of course, is the best that is possible without actually improving the sequential upper bound for the problem. Sometimes one can go one step further, in that one may be able to show that a particular time-processor product is equal to the sequential lower bound for a problem, in which case one says that the algorithm achieves *optimal speedup*. Thus, in general, one hopes to find parallel methods that are fast and achieve optimal or at least linear speedups.

This goal is clouded somewhat, however, in that there is more than one generally accepted model of parallel computation, and each model imposes its own constraints on how close one can come to optimal or linear speedups given certain running times. The specific constraints imposed depend upon which of the two primary classes of parallel models one is working in: the shared-memory model or the network model.

In the shared memory model, commonly referred to as the Parallel Random Access Machine, or PRAM, the processors all have access to a shared random-access memory. This model is further differentiated by the way one resolves concurrent accesses to the shared memory. In the most powerful version, the Concurrent-Read Concurrent-Write (or CRCW) PRAM, one allows for many processors to simultaneously access any memory cell for reading or for writing (with a suitable conflict-resolution rule for concurrent writes). The other versions are more restrictive in that they either disallow simultaneous writes, as in the Concurrent-Read Exclusive-Write (or CREW) PRAM model, or dissallow simultaneous accesses all together, as in the Exclusive-Read Exclusive-Write (or EREW) PRAM model. In any case, the processors act synchronously and all communication takes place by reading and writing in the common memory. The speedup constraints imposed by this class of models, then, is often determined by the structure of one's problem and by the type of memory accesses that are allowed.