

Skeletons for Divide and Conquer Algorithms

Abstract

Algorithmic skeletons intend to simplify parallel programming by providing recurring forms of program structure as predefined components. We present a fully distributed task parallel skeleton for a very general class of divide and conquer algorithms for MIMD machines with distributed memory. This approach is compared to a simple master-worker design. Based on experimental results for different example applications such as Mergesort, the Karatsuba multiplication algorithm and Strassen's algorithm for matrix multiplication, we show that the distributed workpool enables good runtimes and in particular scalability. Moreover, we discuss some implementation aspects for the distributed skeleton, such as the underlying data structures and load balancing strategy, in detail.