Load Bounding for Implicit Parallelism

Gerald Ostheimer
Department of Mathematical and Computational Sciences
University of St. Andrews

Abstract

An obvious problem for parallel computing is how to obtain parallelism from a program specification. A less obvious, but no less important problem can be that of avoiding a 'parallelism explosion'. This situation arises when the amount of parallelism uncovered vastly exceeds the physical parallelism of a machine. The resulting backlog of parallel tasks with its associated storage requirements can lead to the exhaustion of memory resources for algorithms that can run within a moderate amount of space when implemented sequentially. In the context of functional programming, the problem of identifying opportunities for parallelism is typically approached by applying strictness analysis to a program free of special annotations for parallelism. Here we show an approach to the load bounding problem for lazy functional programs based on a compilation scheme which guarantees the load bounding property for the generated code by virtue of its construction, given a perfect load determination function and provided there is no sharing in the source program. We then present a practical approach for estimating the global load. Our strategy is completely distributed and can be expected to scale well to large machine configurations.