Abstract

The precedence constrained task graph scheduling problem on Multiprocessor embedded systems is an important research area. The objectives of this scheduling problem include minimization of the task graph execution time and reduction of overall power consumption. Historically, algorithms have been proposed for a wide range of varying task graph and processor models that primarily favor a static analysis of task graphs to schedule on a target multiprocessor architecture. Among various task graph models, this thesis focuses on the scheduling problem of dynamic task graphs where dynamism in the task graph occurs due to presence of conditional nodes or due to tasks whose execution times vary at run-time. Even though static scheduling methodologies work quite well for task graphs that are mostly static in nature, in the presence of dynamism in task graphs, static scheduling algorithms do not perform equally well. We infer that online adjustments to the schedule can improve the schedule length. This usually happens at the cost of increased scheduling overhead.

This dissertation has two primary objectives. The first is to improve the schedule length of dynamic task graphs by making suitable schedule adjustments at run-time. The second is to use the run-time slacks produced in dynamic task graph schedules for reducing energy requirements of the task graph execution using dynamic voltage scaling techniques. With the above objectives, this thesis approaches the solution by providing a few low overhead online techniques that are often guided by results of offline analysis.

In our first contribution, we present a hybrid scheduling methodology on task graphs (considering negligible communication) and investigate the possibility of code duplication under memory constraints. Subsequently, we have developed online scheduling algorithm for task graphs with communication costs on a target architecture with limited communication channels. We investigate two types of communication models, namely, point-to-point and broadcast models. In the final work, we present an online slack distribution strategy for reducing energy consumption of dynamic task graph schedules. This strategy analyzes task graph schedules in the offline phase to assist the online phase for a better slack distribution. In all the cases, the algorithms aim at maintaining a low complexity, particularly for the online portions.

We study the effectiveness of the proposed methodologies by simulation on a large set of task graphs. The effects of model parameters like number of processors, storage memory, task dynamism, etc, are studied with various experiments and the scope of the proposed techniques is identified. We conclude that the proposed multiprocessor scheduling techniques can provide significant improvements in many situations.

Keywords: Multiprocessor Scheduling, Precedence Constraint, Dynamic Task Graphs, Online Scheduling, Conditional Task Graphs, Slack Distribution