Efficient Method for Periodic Task Scheduling with Storage Requirement Minimisation

Karine Deschinkel, Sid-Ahmed-Ali Touati
University of Versailles Saint-Quentin en Yvelines
45, avenue des Etats-Unis 78035 Versailles, France

Keywords: *periodic task scheduling, storage minimisation, heuristics, linear assignment*

This article addresses the problem of periodic storage optimisation in cyclic data dependence graphs (DDG), which is for instance applied in the practical problem of periodic register allocation for innermost loops on modern Instruction Level Parallelism (ILP) processors. The massive introduction of ILP processors since the last two decades makes us re-think new ways of optimising register/storage requirement in assembly codes before starting the instruction scheduling process under resource constraints. In such processors, instructions are executed in parallel thanks to the existence of multiple small computation units (adders, multipliers, load-store units, etc.). The exploitation of this new fine grain parallelism (at the assembly code level) asks to completely revisit the old classical problem of register allocation initially designed for sequential processors. Nowadays, register allocation has not only to minimise the storage requirement, but has also to take care of parallelism and total schedule time. In this paper, we do not assume any resource constraints (except storage requirement); Our aim is to analyse the trade-off between memory (register pressure) and parallelism in a periodic task scheduling problem. Note that this problem is abstract enough to be considered in other scheduling disciplines that worry about conjoint storage and time optimisation in repetitive tasks scheduling (manufacturing, transport, networking, etc.).

Existing techniques in this field usually apply a periodic instruction scheduling under resource constraints that is sensitive to register/storage requirement. Therefore a great amount of work tries to schedule the instructions of a loop (under resource and time constraints) such that the resulting code does not use more than R values simultaneously alive. Usually they look for a schedule that minimises the storage requirement under a fixed scheduling period. In this paper, we satisfy register/storage constraints early before instruction scheduling under resource constraints: we directly handle and modify the DDG in order to fix the storage requirement of any further subsequent periodic scheduling pass while taking care of not altering parallelism exploitation if possible.

In a previous paper we proposed an exact integer linear model for solving the problem of periodic scheduling with storage minimisation based on a theoretical approach (reuse graphs). Storage allocation is expressed in terms of reuse edges and reuse distances to model the fact that two tasks use the same storage location. Since computing an optimal periodic storage allocation is intractable in large data dependence graphs (larger than 12 nodes for instance) with a classical Branch and Bound method, we present here an efficient heuristic called SIRALINA . Our heuristic proceeds in two steps. A first optimal step provides scheduling variables and allows computing the potential reuse distances if the corresponding reuse edge is added. Then a second step solves a linear assignment problem using the Hungarian method in order to select the appropriate reuse edges.
Our practical experiments on many DDGs show that SIRALINA provides nearly optimal results for all periods in a satisfactory (fast) processing time. Its efficiency, on small problem instances, could be validated with respect to optimal solutions obtained from the exact integer linear model. We also present numerical results on large instances that could not be solved to optimality by an exact method. Consequently, SIRALINA is under inclusion inside a compiler for embedded systems.