Graph-based Cluster Assignment for VLIW Architectures

Alexander Jordan
Academic Advisor: Andreas Knoll
Institute of Computer Languages, Vienna University of Technology
ajordan@complang.tuwien.ac.at

Background

What is a clustered architecture? A superscalar machine that divides its register bank into pieces, of which each is connected only to a subset of the available functional units (FUs), is said to be clustered. This allows for a higher number of FUs, thus increasing instruction-level parallelism (ILP), while keeping the connection logic simple.

How does clustering impact compilers? The job of the compiler gets more complicated when it also needs to make clustering decisions. Apart from evenly utilizing clusters it also needs to transfer values between them.

What cluster assignment approaches are there? Since there are strong interdependencies between cluster assignment and instruction scheduling, which manifest in phase ordering issues, the best results are achieved when both passes are integrated. Due to its complexity, optimal cluster assignment is infeasible, therefore heuristically pre-assigning clusters before scheduling is the most common approach.

Motivation

As long as it remains infeasible for production compilers to integrate register allocation, scheduling and cluster assignment, there is need for efficient cluster assignment heuristics. Examples of the latter are: partial component clustering (PCC) ([1]) and the cluster assignment algorithm proposed by Lapinskii et al. ([2]), which both incorporate an assignment and iterative improvement stage.

Building on these results, the purpose of our work is threefold: we want to:

1. Define a concise model for the cluster assignment task,
2. Compare alternative modeling concepts and cost functions that affect the model, and
3. Apply clustering algorithms from the discrete optimization domain.

Assignment Algorithm

Our cluster assignment algorithm takes as input the directed acyclic graph (DAG) of a basic block after preliminary instruction selection. At this stage, a node in the DAG corresponds to a target instruction that is not yet assigned to a cluster or FU. After performing graph-based clustering, instructions are bound to clusters and completion matrices with register allocation and scheduling.

Step 1: Find paths

Starting at the root node (the node at the bottom of Figure 1(a)), the currently longest path is followed upwards, coloring visited nodes along the way. This is done iteratively until no nodes are left uncolored.

Step 2a: Calculate distances

We regard the discovered paths as a pairwise measure of potential conflict between two graph components (paths) assigned to the same cluster. In a new, reduced graph, components that do not interfere will be clustered together (d = 0), while conflicting components are placed apart proportional to the amount of conflict.

Step 2b: Partition the graph

Now the assignment algorithm needs to find a partitioning of nodes so that intra-cluster conflicts are minimized. This problem corresponds to the well-known min-sum clustering (p being the number of clusters).

Implementation Status

To evaluate our algorithm we use the LLVM (http://www.llvm.org) compiler framework and a prototype-stage code generator for the Texas Instruments TMS320C64x, which has two clusters and four FUs per side. CPLEX is being used to solve the min-sum clustering to optimality. The code is compiled using the LLVM compiler framework and runs on the Texas Instruments TMS320C64x processor.

References


Challenges and Research Directions

The next development steps will refine the machine model for the TMS320C64x and evaluate a number of different distance functions. Comparing the results of our algorithm directly to existing cluster assignment heuristics will serve as a benchmark during this process. Expanding the scheduling region will show the scalability of the LP solver approach. Eventually we want to investigate the integration with instruction scheduling and register spilling.

Acknowledgments

This work is supported by the Austrian Science Fund (Fonds zur Förderung der wissenschaftlichen Forschung) under grant P21842, (Austrian State Conference for Embedded Systems), and the Austrian Science Fund (Fonds zur Förderung der wissenschaftlichen Forschung) under grant P21843.