

Allocation and Scheduling using Linear Programming in the SPIRAL Compiler

Optimizing Compilers : Project pre-proposal

<http://www.ece.cmu.edu/~tstrigko/spiral/> (under construction)

Theodoros Strigkos
strigkos@ece.cmu.edu

Frédéric de Mesmay
fdemesma@ece.cmu.edu

Today's compilers are usually doing register allocation and scheduling in two different passes, with the first pass imposing constraints to the second pass. Different studies [1] have proposed integer linear programming (ILP) to get an optimal solution to those problems but compilation time usually increases exponentially, thus only small codes can be scheduled this way. However, under some conditions [2], the integer linear programming constraints can be relaxed to linear programming (LP) whose response time is polynomial instead of exponential. Our project involves implementing an LP scheduler/allocator within the SPIRAL compiler.

SPIRAL is a Carnegie Mellon University project that generates competitive code entirely autonomously for digital signal processing algorithms and other numerical kernels. When asking for a specific transform, SPIRAL will create a platform specific search space (e.g. taking into account simdization and parallelization) and look for the best implementation. The SPIRAL compiler we will be using takes a simple language as input (loops, additions, multiplications, no pointers) and will be setup to produce fully unrolled SSE code.

Preliminary work by Yevgen Voronenko on Integer Linear Programming has shown potential for significant performance gains for DFTs. However, the algorithm's response latency is not tolerable for large block sizes. We will attempt to reformulate ILP equations to Linear Programming equations, and use an LP solver to compile larger block sizes in reasonable time, while maintaining the performance improvement offered by the ILP algorithm.

References

- [1] C. Chang, C. Chen, and C. King. Using integer linear programming for instruction scheduling and register allocation in multiissue processors, 1997.
- [2] C. H. Gebotys and M. I. Elmasry. A global optimization approach for architectural synthesis. In *ICCAD*, pages 258–261, 1990.