Topic 13: Optimizing compilers - Chapter 5 (Enhancing Fine-Grained Parallelism)

7.11.2015

Martin Orava - xorava01, xorava01@stud.fit.vutbr.cz
Dušan Bezděk - xbezde02, xbezde02@stud.fit.vutbr.cz

Enhancing Fine-Grained Parallelism - Loop Interchange

In this presentation we introduce the topic Enhancing Fine-Grained Parallelism. This topic deals with automatic parallelization of sequential code. Fine-Grained parallelism uses several transformations such as loop interchange, scalar expansion, array renaming, scalar renaming and node splitting.

The main focus of this presentation is in the method loop interchange, which is one of the most useful transformations available for improving program performance. This method is used for switching the order of loops to get parallel loops into the optimal position. The optimal position means making outer loop more vectorization possible in inner loops. The essence of a good loop interchange strategy is to select the right outer loop for sequentiation. We mention safety and profitability of loop interchange as well.

The presentation expands the basic code generation algorithm called Loop interchange. This algorithm is needed to increase the amount of fine-grained parallelism to a level that makes automatic vectorization practical and effective.

References:

- Allen, R., Kennedy, K.: Optimizing Compilers for Modern Architectures. Morgan Kaufmann, 2002