

発表概要

手続き間並列化コンパイラ WPP の試作
— 手続き間 SPMD 化技術 —

佐藤 真 琴†

プログラミングが容易で、広範囲のプログラムに対して高性能が期待できる共有メモリ型並列機 (SMP) が注目を集めている。しかし、最高性能を得るには、手続きにまたがったデータ依存関係がないことやキャッシュ等のマシン特性を最大限に利用する必要があり、人手では多大な労力を要する。このため、逐次プログラムを SMP 向けに自動変換する自動並列化コンパイラへの期待は大きい。我々は、手続き間並列化コンパイラ WPP (Whole Program Parallelizer) において、以下の 3 機能の実現方式を検討し、試作を行なっている: (1) 並列起動等の並列処理コードを手続き全体にまたがって削減する手続き間 SPMD 化, (2) キャッシュ間データ移動を削減するようにループを並列化する手続き間静的アフィニティスケジューリング, (3) キャッシュの効果を考慮した静的実行サイクル数評価を通じて最適なループを並列化する手続き間最適ループ並列化。これらは以下の順序で実現する: まず、並列ループネストを両端に持ち、それらの間にはループネストや分岐等がない逐次部分だけがある SPMD リージョンを手続きにまたがって求める。次に、この中で、キャッシュ間データ移動が発生しない連続ループ群を 1 ノードとするループ類グラフを作成する。最後に、グラフ上の可能なノードの組合せに対する実行サイクル数を評価し、最小値を与えるループ群を並列化する。予備評価では 32 プロセッサで 50 倍の性能向上を得た。

Prototyping of Interprocedural Parallelizing Compiler WPP:
Interprocedural SPMD Region Construction and Parallelization

MAKOTO SATOH†

Symmetric Multi-Processors (SMP) have lately attracted considerable attention because of easy programming and the expectation of getting higher performance for a wide variety of programs. But obtaining the maximum performance by hand requires great efforts because we must make the best use of interprocedural data dependence information and the characteristics of machine resources such as caches. Therefore people are hoping for automatic parallelizing compilers. We have examined and implemented the following three functions on our interprocedural parallelizing compiler WPP (Whole Program Parallelizer): (1) interprocedural SPMD region construction that moves parallel control codes such as thread creations beyond procedure boundaries and reduces them, (2) interprocedural static affinity scheduling that parallelizes such loops as reducing data movement among caches, (3) interprocedural optimal loop parallelization that statically evaluates the execution cycles considering the effect of caches and parallelizes the most appropriate loops. The algorithm of these functions is as follows: first, beyond procedure boundaries, WPP finds SPMD regions that include as their first and last parts two parallel loop nests and as the other parts sequential sections without branches or any loop nests. Second, in each SPMD region, WPP finds groups of contiguous loops among which no inter-cache data movement occurs and makes a loop class graph that deals with each group as a node. Lastly, WPP evaluates the execution cycles of every SPMD region for all possible combinations of nodes in the graph and parallelizes the most appropriate loops that provide the minimum cycle. Preliminary evaluation shows 50 times speed-ups were obtained for 32 processors.

(平成 11 年 6 月 18 日発表)

† 新情報マルチプロセッサコンピューティング日立研究室
RWCP Multiprocessor Computing Hitachi Laboratory