
発表概要

共有メモリ関連命令を生成可能な実装用言語の設計

八 杉 昌 宏^{†,††} 田 畑 悠 介^{††}
小 宮 常 康^{††} 湯 淺 太 一^{††}

並列計算機は共有メモリ型だけを考えてもプロセッサやメモリモデルなどの点で様々なアーキテクチャがある。アーキテクチャの違いを吸収して信頼性・再利用性・実行効率の高いソフトウェアを開発するには、並列処理のための高水準プログラミング言語が有用である。高水準言語コンパイラでは、直接アセンブリコードを生成する代わりに、C言語を実装用言語として利用することでプロセッサに依存するコード生成をCコンパイラに担当させることができる。しかしながら、C言語では、共有メモリに関する不可分操作やメモリアクセス完了順序を直接記述することはできず、ライブラリやasm文などを利用する必要があり、移植性や実行効率を低下させる原因となっていた。本研究ではこれら移植性や実行効率を高めるためのC言語の拡張について検討を行う。性能を犠牲にせずに同じ記述を可能とするためには、適切に選択した機能を組み込みとし、コンパイラにより最適化したコード生成を行うものとした。

Design of an Implementation Language to Generate Code with Shared-Memory Related Instructions

MASAHIRO YASUGI,^{†,††} YUSUKE TABATA,^{††} TSUNEYASU KOMIYA^{††}
and TAIICHI YUASA^{††}

There are various architectures for shared-memory parallel computers in terms of processors and memory models. High-level programming languages for parallel processing are quite useful to develop reliable, reusable and efficient applications on various parallel computers by concealing their architectural difference. Compilers for high-level languages may directly generate assembly code, but they are implemented more easily by employing C language as an implementation language and using C compilers to generate processor-dependent code. In C, however, we cannot directly describe atomic operations and memory orders for the shared memory; we have to use library routines or asm statements, resulting poor portability and lower performance. We present the design of an extended language to C to obtain better portability and performance. In order to enable common description, the language provides carefully-chosen built-in functions and assumes compiler optimization to generate machine-dependent efficient code.

(平成 12 年 8 月 3 日発表)

† 科学技術振興事業団，さががけ研究 21「情報と知」領域グループ

“Information and Human Activity”，PRESTO，Japan Science and Technology Corporation (JST)

†† 京都大学大学院情報学研究科通信情報システム専攻
Department of Communications and Computer Engineering, Graduate School of Informatics, Kyoto University