発表概要

共有メモリ型並列計算機における キャッシュを意識したオブジェクト内レイアウト法

前田 昌樹[†] 鎌田 十三郎[††] 瀧 和男[††]

共有メモリ型並列計算機上でプロセッサ台数に比例した実行速度を得る際に，しばしばプログラマはキャッシュを意識したメモリ配置を行う必要がある．これは，複数のプロセッサが同じキャッシュライン上の異なったデータを参照する際に，一貫性保持の目的で，あるプロセッサによる書き込みが別のプロセッサのキャッシュの無効化を引き起こしメモリアクセス速度が低下してしまうからである．本研究は，変数アクセス情報を利用した，キャッシュミスを抑制するオブジェクト内変数レイアウト法を提案する．共有メモリ型並列計算機を対象としたアプローチとして，我々は，頻繁にアクセスされる変数群をまとめる一方で，頻繁に書き込みが行われる変数に対しては別の領域に配置することでキャッシュコンテンションを抑え，実行効率の向上を目指す．本発表ではレイアウト法として，固定レイアウト法とレイアウト切替法を提案している．固定レイアウト法では，各クラスに対して固定されたレイアウトを利用するのに対し，レイアウト切替法では，計算局面ごとに特化されたレイアウトを準備し，実行時に変数再配置を行う．上記 2 手法の有効性は，実機上のアプリケーションを通しての性能評価により確認する．

Cache-Conscious Field Layouts for Shared-Memory Parallel Processors

Masaki Maeda,[†] Tomio Kamada[††] and Kazuo Taki[††]

To gain scalable speed-up of parallel programs for shared-memory parallel processors, the programmer is often forced to do cache-conscious allocation of memory objects to avoid cache coherency contention. This is because when one processor repeatedly updates a variables on a cache line that other multiple processors use for reading another variable, the readling processors suffers for heavy memory overhead caused by frequent invalidations of that cache line. This presentation proposes two cache-conscious field layout methods for shared-memory parallel computers. These two methods utilize read/write access information of each field of an object, and separately allocate fields with frequent write operations from fields that are frequently read but not written. The first method is fixed splitting that prepares specific field layout for each class. On the other hand, reconfigurable splitting method prepares multiple layouts for one class. Each layout is specialized for a calculation phase of an object, and the object changes its field layout during computation. We evaluate the performance of our layout methods through applications.

(平成 12 年 8 月 3 日発表)

† 神戸大学大学院自然科学研究科
Graduate School of Science and Technology, Kobe University
†† 神戸大学工学部情報知能工学科
Department of Computer and Systems Engineering, Kobe University