

要 旨

行列計算向きストリーム駆動プロセッサの提案

山崎 弘法

近年、マイクロプロセッサとインターネット技術の発展に伴い、big data と呼ばれる大量のデータがネットワーク上で授受されるようになりつつあり、これらを超並列処理するニーズが高まっている。しかし、従来のプロセッサは、以下の律速要因により大量のデータを高速に並列処理することが困難である。1. 各スレッドの起動、停止、同期オーバーヘッド 2. プロセッサに対してメモリが低速であること 3. キャッシュ一貫性保持のオーバーヘッド。

さらに、従来の並列プログラミングは、プログラマがタスクを各スレッドに割り当て、厳密な制御を行う必要があり、大量のデータを高速に処理する並列プログラムを作成することは困難である。

一方、DDP (Data-Driven Processor) はペアとなるデータが揃うと計算を起動するプロセッサであり、ハードウェアレベルでプログラムに内在する細粒度の並列性を実行できるため、並列処理に優れている。しかし、純粋な DFP (Data-Flow Programming) 言語は入力データのデータ構造を定義できず、ベクタ、行列を取り扱うことは困難である。

本研究では、大量のデータを取り扱う際にメモリ・アクセスを必要最小限にし、かつ細粒度並列処理を行うために、巨大な行列計算のためのストリーム処理に着目し、SFP (Stream-Flow Programming) モデルと巨大な行列を格納する SC (Stream Cache) を備えた SDP (Stream-Driven Processor) を提案する。SC は一種の高機能メモリであり、ストリームを生成して DDP に投入し、処理結果を消費する。これにより、SDP ではプロセッサとメモリ間のトラヒックを削減でき、高い並列処理を実現できる。

キーワード 並列処理, 並列プログラミング, データ駆動/ストリーム駆動プロセッサ

Abstract

A Proposal of Stream Driven Processor for Matrix Computation

Hironori YAMASAKI

With the advancement of both microprocessor and internet technologies, demands of massively parallel processing is increasing, especially in the so-called "big data" era. However, the modern multi-/many-core microprocessors could not achieve scalable performance along with the amount of their processing resource because they have the following essential issues; a) invocation, abort, and synchronization overheads of parallel threads, b) performance degradation derived from low memory bandwidth, and c) ensuring overhead of cache coherency. Additionally, modern parallel programming is quite difficult because programmers are responsible for the careful assignment of an appropriate task to each parallel thread and the dexterous control of their concurrent behaviors.

In contrast, the data-driven processor (DDP) architecture is quite suitable to realize fine-grain parallel processing since each operation can be independently executed only when its required operands are available. However, the pure dataflow programming (DFP) language cannot deal with huge data structures such as vector and matrix. This study focuses on the stream processing for huge matrix computation in order to minimize memory access overhead in treating huge data and to perform fine grain parallel processing. This paper discusses a stream flow programming (SFP) model and its stream driven processor (SDP) with a stream cache (SC) storing a part of huge matrix.

The stream cache can operate as a functional memory to produce/consume a stream to/from DDP. Thus, the SDP can reduce traffic between the SC and the DDP so as to perform highly parallel processing.

key words parallel processing, parallel programming, data-driven processor, stream-driven processor.