発表概要

アクセラレータを活用するためのプログラミング環境

中 里 直 人^{†1}

本発表では、GPU や GRAPE-DR などのメニーコア計算機を有効に利用するためのプログラム環境について報告する。メニーコア計算機は、100 個以上のプログラマブルな演算器を集積した LSI からなる計算機である。その性能は単精度では 1 TFLOPS を超え、倍精度では 200 GFLOPS を超える演算性能を持つため、並列計算可能な問題を高速に実行できる可能性がある。一方で、メニーコア計算機は自律的に動作できないため、ホスト計算機を補佐する形で計算を加速するアクセラレータとして利用される。メニーコアアクセラレータは、ホスト計算機のメインメモリとは別のメモリを持つため、必要なデータを明示的にホスト計算機のメモリから転送する必要がある。さらに、その演算器はスクラッチメモリやキャッシュメモリを持つため、どのようにデータを保持するか、どのようにデータを読み込むかの詳細により、演算性能は大きく変化する。それらの利用方法は問題によるため、高性能を引き出すためには、問題ごとにメモリの利用方法を最適化し、加えて、演算器の詳細な動作を最適化する必要がある。我々は、問題を粒子間相互作用の計算に特化することで、最適なコードを生成するコンパイラを開発した。本コンパイラは、利用される変数の性質に応じて、最適なメモリ利用法と演算の詳細を適用し最適化されたコードを生成する。また、本コンパイラでは、通常利用される単精度倍精度だけでなく、四倍精度演算を利用するコードを生成可能である。本コンパイラにより生成された粒子シミュレーションのコードにより、Radeon 4870 GPU で単精度演算で約 500 GFLOPS の演算性能を得た。四倍精度演算のテストケースとして、本コンパイラを高精度が必要な多次元数値積分に適用した。四倍精度に演算性能は、問題の大きさにより Radeon 4870 GPU で 6.4–7.6 GFLOPS、また GRAPE-DR で 2.7–5.5 GFLOPS の性能を得た。これは 1 core の CPU による四倍精度演算性能の数十倍に相当する。以上のように、適した問題ではメニーコアアクセラレータは演算加速器として有効に働く。メニーコアアクセラレータでは、問題に依存した高度な最適化が必要であるが、その上で並列実行可能な問題の種類は限られている。我々の手法を拡張して、問題の種類に応じた最適なメモリの利用法と演算手法を適用する、問題別コンパイラを構築するフレームワークの構想についても報告する。

A Programming Framework for Many-Core Accelerators

NAOHITO NAKASATO^{†1}

We introduce a newly developed compiler for high performance computing using many-core accelerators. A high peak performance of such accelerators attracts researchers who are always demanding faster computers. However, it is difficult to create an efficient implementation of an existing serial program for such accelerators even in the case of massively parallel problems. While existing parallel programming tools force us to program every details of an implementation from loop-level parallelism to 4-vector SIMD operations, our novel approach is that given a compute intensive problem expressed as a nested loop, the compiler only ask us to define a compute kernel inside the inner-most loop. We observe that input variables appeared in the kernel is classified into two types; invariant during the loop and variables updated in each iteration. The compiler let us to specify either type of the inputs so as it pick a predefined optimal way to process them. The compiler successfully generates the fastest code ever for many-particle simulations with the performance of 500 GFLOPS (single precision) on RV770 GPU. Another successful application is the evaluation of a multi-dimensional integral. It runs at a speed of 5–7 GFLOPS (quadruple precision) on both GRAPE-DR and GPU. Finally, we will briefly introduce a possible extension of our compiler system to other problems.

(平成 21 年 10 月 29 日発表)

^{†1} 会津大学
University of Aizu