2014 年 4 月 2 日 情報システム基盤学専攻修士論文中間発表会

ヘテロジニアスコンピューティングでのワークサイズ調整機構に関する研究

所属： 高性能コンピューティング学講座 本多研究室
発表者： 1353017 竹本 拓未
主任指導教員： 本多 弘樹

1 はじめに

近年、CPU や GPU などの異なった種類の演算デバイスを混在させたヘテロジニアス構成のシステムを対象にした並列プログラムの開発環境が普及し始めている。このようなシステムにおいて、計算資源を有効に活用するためには、個々の演算デバイスの演算コア数やメモリ容量などのハードウェア特性やプログラムの特性を把握する必要が不可欠である。さらに、ネットワークを介して複数のコンピュータノードが接続されたマルチノードの複合型計算システムの場合には、ノード間でのデータ転送による時間的オーバーヘッドが顕著に表れる。

このように、ヘテロジニアスコンピューティングを行うために、プログラマはプログラム毎にデバイスの特性に応じたチューニング、および、CPU-GPU 間やノード間の通信速度のオーバーヘッドを隠蔽するための工夫が強いられてしまう問題がある。

そのため本研究では、最適なデバイスに対して、最適な演算スレッド群を割り当てるための機構を開発する。これによって、ヘテロジニアスシステムを対象にした高速に動作するプログラムの開発負担軽減が期待できる。

2 研究背景

ヘテロジニアスシステムを対象としたプログラミング手法として、共通フレームワークである OpenCL[1] を用いた手法が標準化されている。OpenCL を用いることで、1 つのプログラムで複数種類の演算デバイスを使用したプログラムの開発が可能になる。

OpenCL では、演算を行うスレッドのことをワークアイテムと呼称し、1 つのワークアイテムは 1 個の PE（Processing Elements）上で実行される。また、ワークアイテムを任意の個数グループ化したものをワークグループと呼び、1 つのワークグループは 1 個の CU（Compute Units）上で実行される。これらの概要図を図 1 に示す。さらに、演算に使用するデバイス内のワークアイテムの総数をグローバルワークサイズと呼び、1 ワークグループ内のワークアイテムの総数をローカル・ワークサイズと呼ぶ。

ワークグループを構成するワークアイテムの選択やそのローカル・ワークサイズの決定、使用する演算デバイスの決定が、プログラムの実行性能に大きな影響を及ぼす。

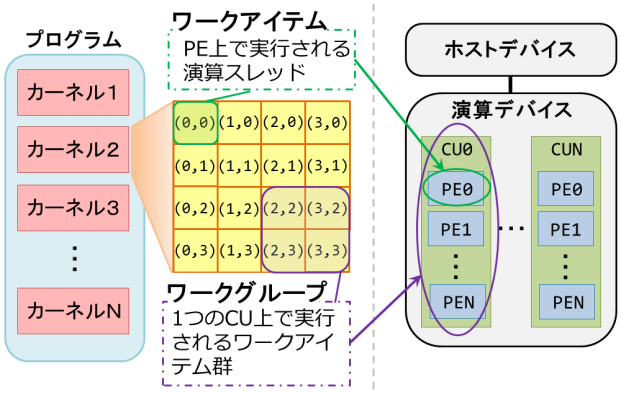

図 1: OpenCL での演算スレッドの実行モデル

関連研究として、GPU プログラムの実行時間を解析する研究が行われている。伊藤らの研究 [2] では、主記憶や GPU メモリ、GPU の演算器間の転送バンド幅と転送遅延の組みをもとに、実行時間を予測する性能モデルを提案している。島田らの研究 [3] では、プログラムの GPU メモリへのアクセス回数や命令数などのパフォーマンスカウンタを用いて実行時間を予測するモデルを作成している。

両研究とも、システムの持つ資源やその性能を取得して実行時間を予測するモデルを作成し、プログラマはその結果をもとにプログラムを手動でチューニングすることを目的としている。

3 提案手法

本研究では、システム内の演算デバイスの持つ資源やその性能などのハードウェア情報と GPU 上で実行させるプログラムの特性の両方をもとに、適切なローカル・ワークサイズのワークグループを自動的に生成し、それらを適切な演算デバイスに適切なグローバル・ワークサイズで割り当てる機構を作成する。

これにより、ヘテロジニアスシステム向けの高速なプログラムの開発負担の軽減を図る。

3.1 グローバル・ワークサイズの調整

ワークアイテムを実行する演算デバイスの決定を行う際に考慮しなければならないのは、各演算デバイスの持つメモリ容量や CU 数、ホストデバイスとの接続形態と、GPU 上で実行させるプログラムのイタレーション数など

のプログラム特性である。これらの情報をもとに、各演算デバイスに割り当てるグローバル・ワークサイズを調整することで、プログラムの高速化を図る。

3.2 ローカル・ワークサイズの調整

ローカル・ワークサイズは、ワークアイテムの選択と各演算デバイスに割り当てるワークグループ数の調整を行うことで調整する。これらの調整では、ワークアイテムを割り当てる演算デバイス内の CU 数や PE 数と、GPU 上で実行させるプログラムの特性を考慮しなければならない。例えば、ワークグループ数が、そのデバイス内の CU 数未満の場合には演算に使用していない CU が生じてしまう。一方で、割り当てたワークグループ数が、そのデバイス内の CU 数を超えた場合にはワークグループの切り替え時間がオーバーヘッドになってしまう。

また、1 つのローカル・ワークサイズに対してそのワークグループ内のワークアイテムの構成には複数の種類が考えられる。例えば、4 × 4 の配列要素の 1 要素を処理するワークアイテムの分割には、1 × 4、2 × 2、4 × 1 の 3 種類がある。分割方法によっては、演算デバイスのキャッシュの空間的局所性などが要因となり、プログラムの実行時間に影響が出ると予想できる。

そのため、ローカル・ワークサイズの調整は演算デバイスの持つ CU 数や PE 数、ワークグループの切り替え時間などを考慮した上で、ワークグループの構成方法も検討する。

4 進捗状況

これまでは、OpenCL や並列・分散システムを有効に利用するための必要知識の習得を行った。

また、表 1 の環境でグローバル・ワークサイズを 16384 で固定し、ローカル・ワークサイズを変化させることで実行時間に影響が出るかどうかを調査した。使用したベンチマークプログラムは行列積 A=B・C を計算するプログラムである。その結果を図 2 に示す。図 2 から、ローカル・ワークサイズを大きくする、つまりワークグループ数を少なくするにつれて実行時間が短くなっていることが分かる。

次に、表 1 の環境でローカル・ワークサイズを固定値にしてワークグループの構成を変化させることで実行時間に影響が出るかどうかを前の実験と同じベンチマークプログラムを用いて調査した。その結果を図 3 に示す。図 3 から、横方向に連続したワークアイテムを 1 ワークグループとして CU に割り当てることで、CU が持つキャッシュを有効に利用でき、実行時間が短くなったと考えられる。

表 1: 実験環境

OS	Scientific Linux 6.4
CPU	Intel Xeon E5-2643
GPU	NVIDIA Tesla K20

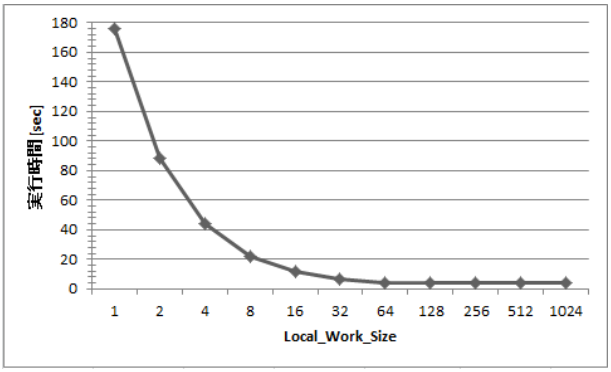

図 2: ローカル・ワークサイズと実行時間の関係

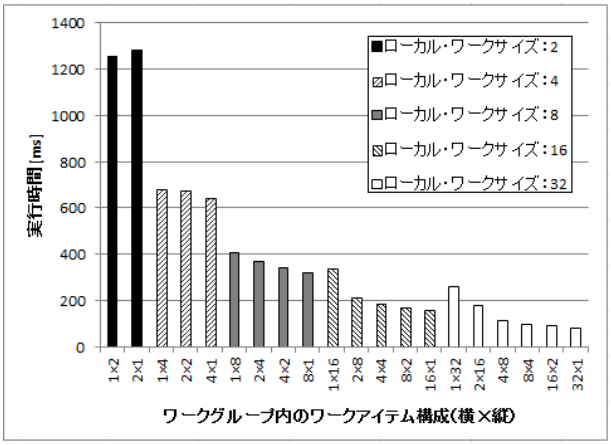

図 3: ワークグループの構成と実行時間の関係

5 今後の方針

今後は、複数種類のベンチマークプログラムを用いて、プログラムごとに最適なワークサイズを求める方法を検討していく。また、ソースコードに対して調整済みのワークサイズを反映させる方法を検討し、実装する。最終的に、OpenCL 以外の GPGPU 向けプログラミング言語との実行時間の比較、評価を行う。

参考文献

[1] The Khronos Group. OpenCL - the open standard for parallel programming of heterogeneous systems -, March 2014. http://www.khronos.org/opencl/.

[2] 伊藤 信悟，伊野 文彦，萩原 兼一．GPGPU アプリケーションの開発を支援するための性能モデル．情報処理学会論文誌コンピューティングシステム，第 48 巻，pp. 235-246，2007.

[3] 島田 大地，遠藤 敏夫，丸山 直也，松岡 聡．OpenCL を用いた異種 GPU における性能特性に応じた最適化．情報処理学会研究報告．計算機アーキテクチャ研究会報告，第 23 巻，pp. 1-7，2010.