

複数GPUのための自動並列化システムの開発

コンピュータサイエンスコース 学籍番号:1411032 成見研究室 大竹聖也

1 はじめに

近年, GPU を本来の画像処理以外の目的で活用する GPGPU (General-Purpose computing on GPU) という技術が注目されている. GPU は CPU と比べて大量のプロセッサを搭載し, 並列処理による高速演算を可能にしているため, 単純な計算においては CPU よりも高い計算性能を示す. 近年, 機械学習など膨大な計算量を要する研究分野が盛んであり, GPU を用いることで計算の高速化を図っている.

また, 研究分野によっては GPU1 基では計算性能が不足し計算時間が膨大になるため, 複数 GPU を用いて計算時間を短縮できる場合がある. そこで, TSUBAME[1] のような複数 GPU からなるスーパーコンピュータが存在し, 研究分野などで活用されている. 複数 GPU を使用すれば, 計算量が少ない場合を除いて GPU が多い方が計算性能が高くなる. しかし, 複数 GPU で計算するためには, ソースコードを複数 GPU が使用できるように変更する必要があり, 追加の CUDA[2] の知識が必要となるなど敷居が高くなってしまふ.

また, 種類の異なる GPU を複数使う場合, 同程度の計算負荷で分割させたときに計算性能の低い GPU はボトルネックになる場合がある. そのため, GPU の資源を最大限に活用するためには, 計算量を計算性能に合わせて変化させる必要があるが, あらかじめその割合を予測するのは難しい.

本研究では, 単一 GPU 向けに書かれたソースコードを複数 GPU 向けのソースコードに変換し GPU ごとの性能差を軽減するための負荷分散を自動で行うことを目的とする.

2 既存研究

2.1 ステンシル計算を対象とした大規模 GPU クラスタ向け自動並列化フレームワーク

野村達雄らは, 流体計算などで利用されるステンシル計算を対象とした大規模 GPU クラスタ向け自動並列化フレームワークを開発した [3]. ユーザーはフレームワークが提供する関数やマクロを使用しソースコードを変換すると GPU で並列化できる. 結果は手動で並列化させた場合より 75% ~125% 程度の計算性能に

なった. 本研究とは, 並列化の対象をステンシル計算に限定し関数やマクロを提供している点で異なる.

2.2 GPU 向け並列計算フレームワークの提案と GA を用いた性能評価

蔵野裕己らは, GPU による並列処理を容易に実現するためのフレームワークを開発し GA (Genetic Algorithm: 遺伝的アルゴリズム) を対象として評価した [4]. 提供された関数で単一 GPU 向けのカーネル関数を定義し GPU で使用するデータや使用する GPU の数も指定する. 本研究とは, 単一 GPU 向けのカーネル関数を使用できる点で一致しているが GPU と送受信するデータを指定する点で異なる.

3 実装したシステム

3.1 システム概要

まず, 既存の単一 GPU を用いるプログラムのソースコードをプログラム `auto_multi.rb` で自作の関数を使用するように変換し, 同時にその関数定義を追加する. 自作した関数は元のプログラム内の CUDA API を複数 GPU で同一の動作をするようにしたものである. この際に, 各 GPU への計算負荷が一定となるように各 GPU で実行するスレッド数を自動的に決定する. 変換後のプログラムは元のプログラムと同一のファイル名になるため, 変換前と同じやり方でコンパイルと実行ができる. また, 自作関数内で環境変数 `[USEGPU0]` と `[USEGPU1]` に GPU の番号を事前に設定しておく必要がある. 図 1 にシステムの概要図を示す.

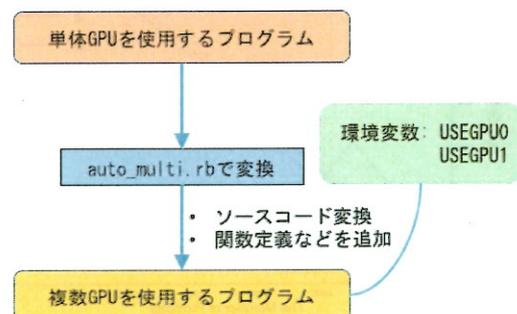


図 1: システム概要図

3.2 計算負荷の最適化

本研究では、計算負荷を最適化するための指標として計算時間を用いた。カーネル関数を実行するたびに実行時間を計測し、記録する。1回目はGPU0に100%、GPU1に0%の計算負荷を与え、2回目はGPU0に90%、GPU1に10%の計算負荷を与える。負荷を10%ずつ変えながら実行し計算時間が前回の計測時よりも増加した場合は前回の計算負荷の比率を使用する。

4 システムの評価

福井大学の古石氏が作成した分子動力学シミュレーションプログラム claret を使用し、手動の場合と本システムの場合で複数 GPU を使用した際の性能を比較して評価する。NaCl イオンに働く力の計算を100回行った時の計算時間から計算速度を算出した。また、使用した3つのGPUのうち GeForce GTX780 を1, GeForce 8800 GT を2, GeForce GTX460 を3と呼称する。

4.1 単一 GPU との比較

1つのGPUと2つのGPUにおける計算性能の差を比較した。2つのGPUについては3通りの組み合わせで計測した(図2)。この実験では選ぶGPUは固定しており、2つのGPUへの負荷をシステムが自動で決定している。2GPUのときはGPU3のみの場合を除いてGPU1よりも計算性能が低下していることがわかった。

4.1.1 手動並列化と比較した場合

手動でGPUの並列化を行った場合と本システムが自動で行った場合を比較した(図3)(グラフのautoが自動, manualが手動の場合の結果)。粒子数が約6000のときまでは手動の場合と自動の場合でほとんど差はないが、それ以降は手動の場合の方が計算性能が高い。同じGPUの組み合わせで比較すると10240粒子のときどちらも約50GFLOPSの差が生じた。

手動で並列化した claret プログラムは各 GPU 単体

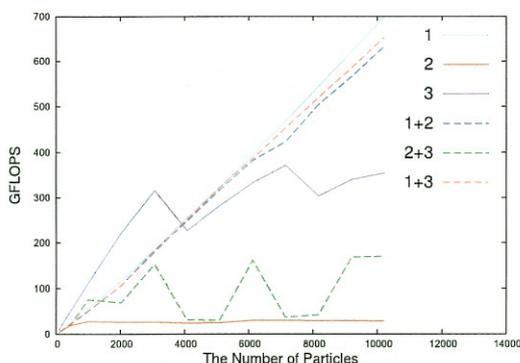


図 2: 単一 GPU との比較

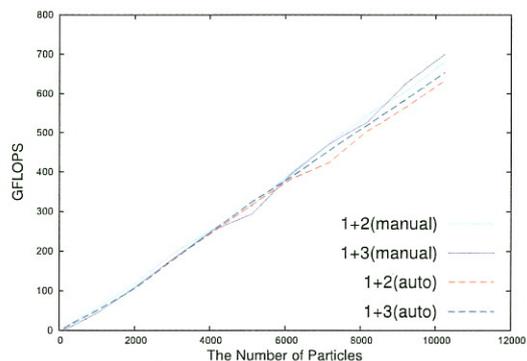


図 3: 手動並列化と自動並列化の比較

の計算時間を基に割合を算出しているため本研究の自動化システムより厳密で、10240粒子のとき手動の場合の割合(全体を100とする)は、GPU1:GPU3=84:16であり、自動の場合の割合は、GPU1:GPU3=90:10であった。

5 まとめと今後の課題

本研究では、単一GPU向けに書かれたソースコードの並列化と負荷分散を自動で行うためのスクリプトを開発しソースコードの変換や自作関数の追加を行った。その結果、自動並列化や負荷分散の機能が追加できたが、並列化による計算性能の向上は実現できなかった。

今後の課題としては、手動による並列化との差が存在するため負荷分散の方法や自動化による処理の増加を抑える方法を考える必要がある。また、現在のシステムではGPUを3つ使うとスレッドが生成されず計算ができないため、その原因を調査し改善する必要がある。claretプログラムに合わせて調整した関数が多いため、より一般的な自動並列化システムを作成すべきである。そのためには、GPUでの計算で使う変数の個数に制限を設けないことやカーネル関数の計算結果を取得するなどを改善しなければならない。

参考文献

- [1] "TSUBAME とは", <http://www.gsic.titech.ac.jp/tsubame>, 2018年1月29日
- [2] "CUDA とは?", http://www.nvidia.co.jp/object/cuda_what_is_jp.html, 2018年1月29日
- [3] 野村達雄, 丸山直也, 遠藤敏夫, 松岡聡 "ステンシル計算を対象とした大規模GPUクラスター向け自動並列化フレームワーク", 情報処理学会研究報告(2010), Vol.2010-HPC-128 No.7
- [4] 蔵野裕己, 三木光範, 吉見真聡, 廣安知之 "GPU向け並列計算フレームワークの提案とGAを用いた性能評価", 情報処理学会研究報告(2011), Vol.2011-HPC-132 No.11