

## 3P094

### GPU による量子化学計算に基づく高速静電ポテンシャル計算の研究

(東大院工\*, 東大生研\*\*)○湯川英宜\*, 平野敏行\*\*, 恒川 直樹\*\*, 佐藤文俊\*\*

#### 【緒言】

GPU は従来 3D グラフィックス処理を行うためのプロセッサであったが, その高度な並列処理能力から近年汎用的な計算への応用がされ始めている. 計算化学の分野においても古典計算に基づく静電ポテンシャル計算[1]や量子化学計算における 4 中心 2 電子積分の計算[2]への応用例が報告されている. タンパク質をはじめとする生体高分子の精密な静電ポテンシャルはその分子の重要な物性となるが, 全電子波動関数から得られる静電ポテンシャル計算は, 対象分子が巨大ゆえに長い計算時間を要する. 本研究では, 量子化学計算に基づく大規模分子の静電ポテンシャル計算を GPU で実行可能なプログラムとして実装し, 静電ポテンシャルの計算時間を大幅に短縮したのでこれを報告する.

#### 【静電ポテンシャルの計算方法】

ある点  $\mathbf{r}$  における静電ポテンシャルは, 原子核および電子からの静電ポテンシャルの和として求められる(式(1)). 原子核からの静電ポテンシャルにおいて計算強度は分子サイズの 1 乗に比例する(式(2)). 一方, 電子からの場合, その計算量は分子サイズの 2 乗に比例するため, 大規模分子では膨大な計算量となる. 本研究では計算量の多い電子からの静電ポテンシャルの高速化に注目した.

$$\phi(\mathbf{r}) = \phi_{nuc}(\mathbf{r}) + \phi_{elec}(\mathbf{r}) \quad \dots(1)$$

$$\phi_{nuc}(\mathbf{r}) = \sum_{A=1}^M \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} \quad \dots(2)$$

$$\phi_{elec}(\mathbf{r}) = -\int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' = -\sum_{pq} P_{pq} \int \frac{g_p(\mathbf{r}')g_q(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' \quad \dots(3)$$

#### 【GPU による静電ポテンシャルの計算方法】

GPU の性能を引き出すには, 計算結果を保持するメモリ領域の確保とその結果の転送方法, CPU からの計算に必要なデータの転送にかかるコスト, GPU での並列処理方法, GPU 特有の高速にアクセス可能なメモリの使用法など GPU の内部構造と並列化を十分に考慮したプログラミングが必要である. 本研究では, GPU の性能を十分に引き出すため, 以下に示す内容のコーディングを行った.

#### ①ホスト-GPU 間のメモリ転送量の削減

GPU でのプログラミングでは演算内容を記すだけではなく, 計算結果を残しておくための領域の確保や転送なども指定する必要がある. 本研究では, はじめに計算結果のための領域を確保し, 最後に計算結果をまとめて CPU 側へ転送するだけで良い内容にするようにした. 結果として, メモリ領域の初期化や CPU-GPU 間でのデータ転送による遅延を極限に抑えることができた.

また GPU において計算に必要な情報をホストから転送する必要がある. ホスト GPU 間のデータの転送速度はホスト内・GPU 内に比べて遅く, 頻繁に行くとプログラムのボトルネックとなる. 本研究では, 基底関数にガウス関数を用いており, GPU へ基底関数を転送する前にあらかじめ縮約ガウス関数の組み合わせをスクリーニングし, スクリーニング後のデータのみを転送した. また原始ガウスの情報をまとめて転送し, GPU 内部で各原子ガウス関数の組み合わせでスクリーニングをするという構造にすることでデータ転送量を抑えることができた.

②グリッド領域を3次元配列から1次元配列へ変換

GPU側で3次元グリッドをそのまま並列処理するプログラムの記述は難しい。本研究では各グリッドを1次元配列で指定できるようにすることで円滑な並列処理を行えるようにした。またグリッドの総数はGPUのプロセッサ数を上回るため、GPUが一度に並列処理できるだけのグリッド数を繰り返し並列に計算できるようなループ構造を作った。

③GPU上の高速アクセス可能なメモリの利用

GPU内のマルチプロセッサ内にはシェアードメモリと呼ばれる16KBの高速アクセス可能なメモリが存在する。繰り返し使う変数などはこのメモリに入れることでより高速な演算処理ができる。全てのグリッド点の計算において電子密度行列要素 $P_{pq}$ との積を求めることになるため、電子密度行列要素 $P_{pq}$ をシェアードメモリに入れることで高速化を図った。

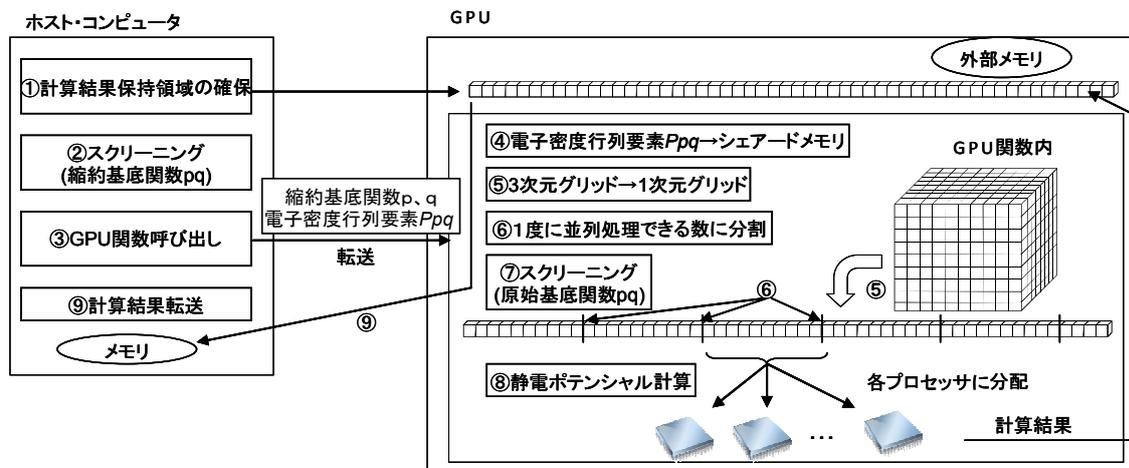


図1 GPUを用いた場合の静電ポテンシャルの計算手順

【結果】

図2に本研究の方法(A)と、対照として全てCPUで計算した場合(B)における軌道数と計算時間の関係を示した。表1には同結果の各軌道数における計算時間と(A)と(B)の計算時間の比を示した。CPUはOpteron Dual Core 2.0GHz, GPUはTesla C870(理論ピーク性能500GFLOPS)を用いた。図1の両対数グラフの傾きから(A)および(B)のサイズ依存性はそれぞれ2.1, 2.4と求められた。表1から、インスリン(4443軌道)の計算時間は、25時間(CPUのみによる計算)から29分(GPUによる計算)へと大幅に削減されたことがわかった。使用したCPUは2.0GHzのDual Coreであるため、GPUの実行性能は約200GFLOPS(=4.0GFLOPS×52)もの性能を出せたことになる。

【参考文献】

[1] C. I. Rodrigues, D. J. Hardy, J. E. Stone, K. Schulten, W. W. Hwu, *Proceedings of the 2008 conference on Computing frontiers*, 273 (2008).  
 [2] K. Yasuda, *J. Comput. Chem.*, **29**, 334 (2008).

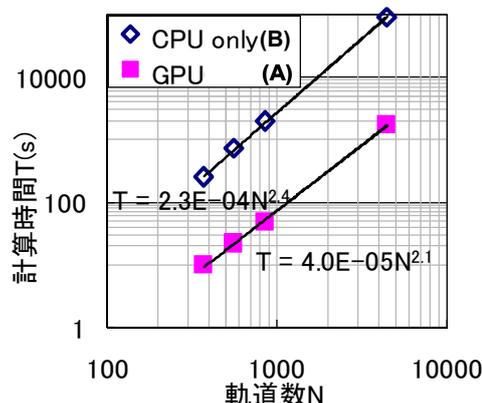


図2 軌道数と計算時間の関係

表1 CPUのみによる計算時間(s)とGPUを使用した際の計算時間(s)とその比

軌道数	cpu(s)	gpu(s)	cpu/gpu
357	251	10	25
525	732	22	33
783	1955	49	40
4443	91558	1750	52