

## GPGPU によるエネルギー分布関数計算の高速化

(金沢大院・自然)塩釜 拓仁, 齋藤 大明, 川口 一朋, 長尾 秀実

GPGPU acceleration for calculation of energy distribution function

(Kanazawa Univ.)Takuhiro Shioyama, Hiroaki Saito,  
Kazutomo Kawaguchi, Hidemi Nagao

## 1 はじめに

近年の計算機性能の向上によりタンパク質、その複合体、脂質膜などの大規模な生体系に対してのコンピュータシミュレーションが広く行われるようになってきた。しかし大規模系の計算では粒子数が大きいため計算コストも高い。したがってシミュレーションを高速化する手法やハードウェアの開発が続けられている。現在開発されている次世代スパコンとして、GPU をアクセラレータとして使ったスーパーコンピュータが挙げられる。したがって CUDA や OpenCL, OpenACC などの GPU を利用するための API の開発が進められている。分子シミュレーションの中でも自由エネルギーの計算は特に計算コストが高い。自由エネルギーを求める手法に自由エネルギー摂動法や熱力学積分法がある。これらの方法では多数の中間状態を作成しそれぞれに自由エネルギーを求めるための計算を行わなければならない。そこで松林らは溶液の分布関数理論に着目し自由エネルギーを低コスト、高精度で計算するエネルギー表示法を開発した [1-3]。この方法では溶液系のエネルギー分布関数を求める。このエネルギー分布関数を求める際に粒子間相互作用を計算する必要がある。そのため粒子数の二乗に比例して計算コストが増大する。そこで本研究では相互作用計算を CUDA Fortran を用いて高速化し、性能の評価を行う。

## 2 エネルギー表示法

ここではある分子を溶質として溶媒に溶かす場合を考える。その時の溶質-溶媒間相互作用の 2 体ポテンシャル  $v$  を予め設定しておく必要がある。 $v$  は溶質配置  $\psi$  と溶媒配置  $x$  の関数である。 $x_i$  を  $i$  番目の溶媒分子の座標とすると、溶質の周りの溶媒の瞬間的なエネルギー分布関数  $\hat{\rho}(\epsilon)$  は

$$\hat{\rho}(\epsilon) = \sum_i^N \delta(v(\psi, \mathbf{x}_i) - \epsilon) \quad (1)$$

で定義される。(1) 式中の  $\sum$  は  $N$  個の全溶媒分子について和を取る。ここで溶質-溶媒間相互作用がある系の  $\hat{\rho}(\epsilon)$  の統計平均を  $\langle \dots \rangle$ 、無い場合の統計平均を  $\langle \dots \rangle_0$  と表す。するとそれぞれの系の平均分布  $\rho(\epsilon)$ 、 $\rho_0(\epsilon)$  は

$$\rho(\epsilon) = \langle \hat{\rho}(\epsilon) \rangle, \rho_0(\epsilon) = \langle \hat{\rho}(\epsilon) \rangle_0 \quad (2)$$

と書かれる。そしてエネルギー表示の密度汎関数理論を用いて定式化を行えば溶媒和自由エネルギー  $\Delta\mu$  は

$$\Delta\mu = \int d\epsilon \rho(\epsilon) - k_B T \int d\epsilon [(\rho(\epsilon) - \rho_0(\epsilon)) - \rho(\epsilon) \log\left(\frac{\rho(\epsilon)}{\rho_0(\epsilon)}\right) - \alpha(\epsilon)F(\epsilon) + (1 - \alpha(\epsilon))F_0(\epsilon)(\rho(\epsilon) - \rho_0(\epsilon))] \quad (3)$$

ここで  $\alpha(\epsilon)$  は重み関数である。 $F(\epsilon)$ 、 $F_0(\epsilon)$  はシミュレーション上  $\rho, \rho_0$  を厳密に求めることができないために導入された関数である。本研究ではエネルギー表示法の最初の段階であるエネルギー分布関数  $\rho(\epsilon)$ 、 $\rho_0(\epsilon)$  を求めるプログラムについて高速化を施す。対象とする系は水分子 647 個を溶媒として水分子 1 個をその中に溶かすもの考える。水分子のモデルは TIP3P である。

## 3 高速化手法

$\rho$ 、 $\rho_0$  を求める計算ではエワルドの方法によるクーロン相互作用の計算を含む。その部分が計算時間の 90% を占めている。そのため、この部分に対して CUDA Fortran を使って高速化を施す。GPU はグリッド、

ブロック、スレッドといった階層構造を持っている。最小の単位はスレッドでありスレッドごとに同じプログラムが実行され、データを GPU 側のメモリとホストの CPU 側のメモリと通信することによって計算結果を得る。従ってこの場合 do ループを GPU 上で総スレッド数に分割し、求めたデータをホスト側へコピーするといった手法をとる。本研究では NVIDIA 社の Tesla C1060 を搭載した計算機、Tesla C2075 を搭載した計算機で性能評価を行った。Tesla C1060 のアーキテクチャは GT200、Tesla C2075 のアーキテクチャは Fermi である。なお MPI,cuda 並列いずれの場合でも通信に掛かる時間がボトルネックになりやすい。MPI の場合ノード間の通信はイーサネットを介して行われるため低速である。GPU の場合、ホスト GPU 間のメモリ上のデータのコピーは、PCIExpress を介して行われるため低速である。従ってホストデバイス間のコピーは極力避けるべきである。なるべく通信の回数を減らすべきであり、通信されるデータの量も減らすべきである。したがって do ループの構造を変更しカーネル関数の呼び出しを最小限にするよう最適化を施した。並列化する手法としてはどちらも粒子分割であり、単純に do ループの回数を全 cpu 数またはスレッド数で分割した。

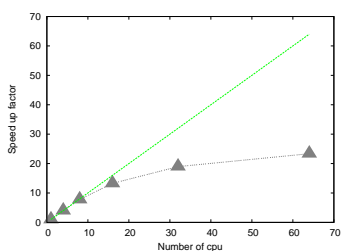


図 1: 相互作用計算の高速化率。図中の直線は高速化率の理論値であり実際の値は  $\rho$  で示した。

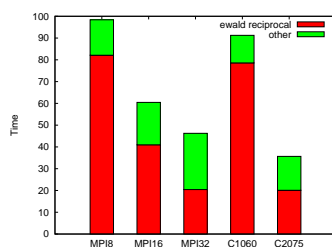


図 2:  $\rho$  を計算するプログラムの高速化。MPI18,16,32 は MPI 並列を 8, 16, 32 並列化した結果、C1060,C2075 は各々のアーキテクチャを持った GPU を使用した結果を示す。

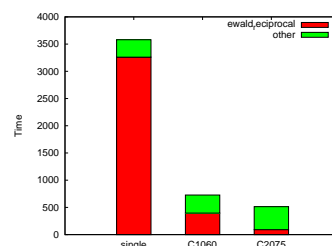


図 3:  $\rho_0$  を計算するプログラムの高速化。single は 1core での計算時間の内訳、C1060,C2075 はそれぞれのアーキテクチャを持った GPU での計算時間の内訳である。

## 4 結果

最初に MPI による並列化に対する結果を示す。高速化率を  $\text{高速化率} = T_1/T_n$  で評価する。 $T_1$  は 1 プロセッサでの計算時間  $T_n$  は n プロセッサでの計算時間である。図 1 は  $\rho(\epsilon)$  を計算するプログラムにおけるプロセッサ数に対する計算時間と高速化率である。次に GPU を用いた結果について説明する。図 2 は  $\rho(\epsilon)$ 、図 3 は  $\rho_0(\epsilon)$  を計算するプログラムについての計算時間の内訳について MPI 並列と CUDA による並列を比較した結果である。図中の赤で示した部分はエワルドの方法によるクーロン相互作用の逆格子空間での計算時間である。緑で示した部分はその他の部分である。図 2 の結果から、MPI の並列数が増えるとその他の部分の計算時間が増大している事がわかる。これは並列数が増えることにより通信にかかる時間が増加するためである。図 2、図 3 の結果から GPU のアーキテクチャの違いが大幅に計算性能を向上させる事がわかる。

## 5 まとめ

MPI のみを使って高速化した結果から並列度を上げててもかなり早い段階で高速化率は飽和してしまうことがわかった。これは通信のオーバーヘッドが大きく影響しているからだと考えられる。GPU を使って高速化した場合について、アーキテクチャの違いが速度に大きく影響した結果となったが、 $\rho(\epsilon)$  の計算は約 60 倍、 $\rho_0(\epsilon)$  の計算については約 42 倍の高速化に成功した。

## 参考文献

- [1] N. Matsubayashi and M. Nakahara J. Chem. Phys. 113, 6070 (2000)
- [2] N. Matsubayashi and M. Nakahara J. Chem. Phys. 117, 3605 (2002)
- [3] N. Matsubayashi and M. Nakahara J. Chem. Phys. 119, 9689 (2003)