

15-NA15

分子性液体の積分方程式理論による大規模生体分子系における高速な溶媒和自由エネルギー計算プログラムの開発

丸山 豊（理化学研究所 計算科学研究機構）

生体系のメカニズム解析や創薬への応用を目指して、大規模な生体分子系における溶媒和自由エネルギー計算の高速化を行なうのが本研究の目的である。そのために生体分子の周りの水を統計力学的に取り扱う事のできる分子性液体の積分方程式理論のプログラムのベクトル化及びマルチ GPU 化を行なった。さらにプログラムの精度を検証するために蛋白質-蛋白質相互作用の計算を行い、実験との比較を行なった。

1. 共同研究に関する情報

(1) 共同研究を実施した拠点名

東北大学

東京工業大学

(2) 共同研究分野

- 超大規模数値計算系応用分野
- 超大規模データ処理系応用分野
- 超大容量ネットワーク技術分野
- 超大規模情報システム関連研究分野

(3) 参加研究者の役割分担

丸山 豊(理化学研究所 計算科学研究機構):

研究総括、プログラム作成、計算科学

額田 彰(東京工業大学):

プログラム作成、大規模並列処理、GPU コンピューティング、3D-FFT の高速化並列化、計算機科学

2. 研究の目的と意義

生体系のメカニズム解析や創薬等への応用を目指して、大規模な生体分子系における溶媒和自由エネルギー計算の高速化を行なう事が本研究の目的である。

重要な化学反応や生体反応の多くは水中で起きている。また、蛋白質などの生体分子が構造を形成し機能を発揮するためには水が必要不可欠である。つまり蛋白質の自発的な構造形成のメカニズムを明らかにし、創薬などで薬効分子との結合を考慮するためには、周りの水を考慮する必要がある。

この水の効果は溶媒和自由エネルギーという物理量で表わす事ができる。通常分子動力学法 (MD) シミュレーションでは、あらわに水を入れて取り扱う事ができるが、溶媒和自由エネルギーや部分モル体積などの物理量を計算するのは計算コストが大きく難しい。また、蛋白質の揺らぎを考えると多数のコンフォメーションにおける計算が必要となり、MD における計算はより困難となっている。

このように重要な生体反応の多くは膜蛋白質で起こっており、生体膜も含めた水中の大きな生体分子系における高速で正確な溶媒和自由エネルギーの計算が求められている。これに対応する一つの解として積分方程式理論によるアプローチが挙げられる。分子性液体の積分方程式理論 (3D-RISM) は生体分子の周りの水を統計力学的に取り扱う事ができる。分子の周りの水の分布関数を計算し、それにより溶媒和自由エネルギーなどの物理量を求められる。計算時間も GPU を用いて数秒から数分 (タンパク質の大きさに依存する) で計算可能である。ただ、GPU は一般に CPU よりもメモリが小さいために GPU 単体で計算可能なタンパク質の大きさに制限がある。しかし、この制限はマルチ GPU 化によって克服可能である。

溶媒和自由エネルギーは、例えば薬効分子と蛋白質の結合強度などを正確に見積もるために重要な物理量である。今までの 3D-RISM 理論では、計算値が溶質分子の大きさに依存して実験値から乖離していく傾向がある事が知られていた。これ

は、理想流体を出発点として理論を構築しているためである事が近年明らかになった。そこで我々は剛体球を参照系とする事で正確な値を算出する溶媒和自由エネルギー表式(RMDFT)を導いた。3D-RISM 理論と RMDFT 表式を組み合わせる事でアミノ酸側鎖分子の溶媒和自由エネルギーが実験値と良く一致する事を示した[1]。

ベクトル型スパコンやマルチ GPU に対応した 3D-RISM と RMDFT のプログラムを作成し、大規模な生体分子系における溶媒和自由エネルギー計算の高速化を行なった。

さらに理論の精度の検証を行なうために、作成したプログラムを用いて蛋白質-蛋白質間の相互作用エネルギーを計算した。

[1] T. Sumi, A. Mitsukake, and Y. Maruyama, J. Comput. Chem. 36, 1359-1369 (2015).

3. 当拠点公募型共同研究として実施した意義

大規模な生体系における高速な溶媒和自由エネルギー計算を目指して、ベクトルマシンや GPU を搭載したクラスター上でのプログラム開発を行なった。プログラムの開発を効率的に行なうためにアプリケーション側(丸山)とシステム側(額田)の共同研究を実施した。アプリケーション側の丸山は 3D-RISM プログラムの GPU 化[1]や超並列化(京スパコン)[2]に取り組んできた経験がある。一方、システム側の額田は地球シミュレータや SGI Altix 3800、IBM Blue Gene/L など多様なシステムで FFT の高速化に取り組んできた実績がある。[3-5]

当該プログラムは 3 次元 FFT を多用し、計算時間における割合が大きい。そこで、額田がベクトルマシン用の FFT のチューニングを行なった。また、当該プログラムの FFT 以外の部分もメモリ律速であるため、スカラーマシンよりもメモリバンド幅の大きいベクトルマシンの方が高速化に適している。額田が作成した FFT プログラムを結合し、さらに他の部分のベクトル化を行なった。

一方でベクトルマシンは数が少なく一般のユーザーが利用するのは難しい。そこで、開発したプロ

グラムを広く利用してもらう事を考慮し GPU クラスタにおける開発も行なった。GPU の欠点はメモリ量が少なく単体では大規模計算に向かないためにマルチ GPU 化が必要であった。しかし、丸山がシングル GPU での高速化と超並列マシンの並列プログラムの実績があり、MPI によるマルチ GPU 用プログラムの開発は比較的容易に行なえた。

以上のようにベクトルマシンと GPU クラスタの 2 つのタイプのコンピュータにおける開発をアプリケーション側とシステム側で効率的に行なうために、公募型共同研究を利用した。

[1] Y. Maruyama and F. Hirata, J. Chem. Theory Comput. 8, 3015-3021 (2012).

[2] Y. Maruyama, N. Yoshida, H. Tadano, D. Takahashi, M. Sato and F. Hirata J. Comput. Chem. 35, 1347-1355 (2014).

[3] 額田彰, 西田晃, 「地球シミュレータを用いた高性能 2 次元 FFT」, 2007 年先進的計算基盤システムシンポジウム論文集, 137-144, 情報処理学会, (2007).

[4] A. Nukada, D. Takahashi, R. Suda, and A. Nishida. In Proceedings of the Third International Conference on High Performance Computing and Communications (HPCC 2007), Lecture Notes in Computer Science 4782, 396-407, Springer, (2007).

[5] A. Nukada, Y. Hourai, A. Nishida and Y. Akiyama. In Proceedings of the Fifth International Symposium on Parallel and Distributed Processing and Applications (ISPA07), Lecture Notes in Computer Science 4742, 958-969, Springer, (2007).

4. 前年度までに得られた研究成果の概要

5. 今年度の研究成果の詳細

5.1 3D-RISM 及び RMDFT

ここでは 3D-RISM 理論及び RMDFT 表式の詳細には触れずに主に計算手順について説明する。

3D-RISM 理論は 3D-RISM 方程式

$$h_{\gamma}^{uv}(\mathbf{r}) = \sum_{\gamma'} c_{\gamma'}^{uv}(\mathbf{r}) * (\omega_{\gamma'}^{vv}(r) + \rho^v h_{\gamma'}^{vv}(r))$$

とクロージャー方程式 (PLHNC 方程式)

$$h_{\gamma}^{uv}(\mathbf{r}) = \begin{cases} \exp(\chi) - 1 & \text{for } \chi \leq 0 \\ \chi & \text{for } \chi > 0 \end{cases}$$

$$\chi = -\beta u_{\gamma}^{uv}(\mathbf{r}) + h_{\gamma}^{uv}(\mathbf{r}) - c_{\gamma}^{uv}(\mathbf{r})$$

の 2 つから成り、それぞれ交互に計算して収束させていく。ここで $h_{\gamma}^{uv}(\mathbf{r})$ 及び $c_{\gamma}^{uv}(\mathbf{r})$ は、全相関関数及び直接相関関数、 $\omega_{\gamma}^{vv}(r)$ は分子内相関関数、 γ 及び γ' は溶媒分子の相互作用点のインデックス、 $h_{\gamma}^{vv}(r)$ はバルクの溶媒の全相関関数、 ρ_v は溶媒の数密度、 $u_{\gamma}^{uv}(\mathbf{r})$ は相互作用ポテンシャル、 $\beta = 1/(k_B T)$ は逆温度である。また * は畳み込み積分を表す。この畳み込み積分により 3D-RISM 方程式はフーリエ空間で積の形で表されるため、各要素を 3 次元フーリエ変換してフーリエ空間で計算する。一方、クロージャー方程式は実空間で計算を行なう。このため、3D-RISM 理論を解くためには、3 次元 FFT を繰り返して実行し、収束させていく必要がある。

物理量の一つである溶媒和自由エネルギーは、今まで Singer-Chandler (SC) 表式

$$\Delta\mu = \rho k_B T \sum_{\alpha} \int_{V_{cell}} dr \left[\frac{1}{2} (h_{\gamma}(\mathbf{r}))^2 \Theta(-h_{\gamma}(\mathbf{r})) - c_{\gamma}(\mathbf{r}) - \frac{1}{2} h_{\gamma}(\mathbf{r}) c_{\gamma}(\mathbf{r}) \right].$$

により計算されてきた。しかしこの式は溶質の原子数に依存して絶対値を過大評価してしまう。この問題は 1 次元の RISM 理論でも指摘されていた。その一方で自由エネルギー差はおおよそ正しい値を与える事も知られていた。そのため、溶質の原子数が増えない現象、例えば蛋白質の折りたたみ問題に適用し、自由エネルギー差を論点とする場合には大きな問題とならなかった。

この溶媒和自由エネルギーの過大評価の問題を改善するために経験的な修正を加える式がいくつか提案されている。これらの式は低分子での結果を元に修正を行なうものであり、蛋白質などの巨

大な生体分子にどこまで適用できるかは不明である。我々はこの問題点を本質的に改善するために、密度汎関数理論に基づき、剛体球を参照系とする高精度の新しい溶媒和自由エネルギー表式を導いた。この式は

$$\Delta\mu = -\frac{1}{\beta} \sum_{\gamma} \int d\mathbf{r} \rho_{\gamma} h_{\gamma}(\mathbf{r}) - \int d\mathbf{r} [\rho_{\gamma} h_{\gamma}(\mathbf{r}) \{ \int d\mathbf{r}' W_{oo}(|\mathbf{r}-\mathbf{r}'|) \times [\rho_o h_o(\mathbf{r}') f'_{HS}(\rho_o h_o(\mathbf{r}')) - \rho_o f'_{HS}(\rho_o)] + \int d\mathbf{r}' W_{oo}(|r|) \rho_o f'_{HS}(\rho_o) \} - \rho_o f'_{HS}(\rho_o) \int d\mathbf{r}' W_{oo}(|r|) \rho_o] + \frac{\rho_o}{\beta} \sum_{\gamma} \sum_{\gamma'} \int d\mathbf{r} d\mathbf{r}' C_{\gamma\gamma'}(|\mathbf{r}-\mathbf{r}'|) \rho_{\gamma} h_{\gamma}(\mathbf{r}') + \frac{1}{2\beta} \sum_{\gamma} \sum_{\gamma'} \int d\mathbf{r} d\mathbf{r}' C_{\gamma\gamma'}(|\mathbf{r}-\mathbf{r}'|) \rho_{\gamma} h_{\gamma}(\mathbf{r}) \rho_{\gamma'} h_{\gamma'}(\mathbf{r}')$$

と表される。ここで $f'_{HS}(\rho)$ は剛体球の 1 粒子当たりの余剰エネルギーの 1 次微分、 $C_{\gamma\gamma'}(\mathbf{r})$ は剛体球の効果を含んだバルクの溶媒の直接相関関数、 $W_{\gamma\gamma'}(r)$ はバルクの剛体球の 2 次の直接相関関数を表す。この表式により溶媒和自由エネルギーの絶対値がより正確に求められるようになった。

具体的な計算手順は

- (1) 入力ファイルの読み込み
- (2) 計算セルの各グリッド点におけるポテンシャルの計算
- (3) クロージャー方程式の計算
- (4) 3 次元 FFT 正変換
- (5) 3D-RISM 方程式の計算
- (6) 3 次元 FFT 逆変換
- (7) 収束判定をして収束していれば (9) へ
- (8) 収束加速法により新しい予測分布を求めて (3) へ
- (9) 物理量の計算

となる。この計算は (3) から (8) の繰り返し部分が最も重くなっている。また 3 次元 FFT はその中でも約 40% (Intel Xeon で OpenMP による計算した場合) を占める。(9) の物理量の計算で SC や RMDFT の式により溶媒和自由エネルギー等を求める。

また、MPI で並列化する場合、3D-RISM 方程式及びクロージャー方程式の計算は完全に並列化され通信を行なわない。通信を行なうのは 3 次元 FFT

と収束判定であり、実行時間の割合は 3 次元 FFT の部分が多い。また 3D-RISM の結果を用いて溶媒和自由エネルギーの計算を行なう RMDFT は、複数回の 3 次元 FFT を必要とする。

以上のように 3 次元 FFT は溶媒和自由エネルギー計算において必要不可欠なものであり、また、高速化が望まれている。我々は SX-ACE 及び TSUBAME2.5 で 3 次元 FFT の高速化や並列化に取り組んだ。特に大規模な生体系の計算を念頭に置いて $512 \times 512 \times 512$ の高速な 3 次元 FFT を目指した。

5.2 SX-ACE

NVIDIA 社製 GPU と SX-ACE のベクトルプロセッサではある程度の類似性があると考えられる。メモリ階層としては、共に高いメモリバンド幅を備える外部メモリと、内部メモリとして高速にアクセス可能だが容量の限られるキャッシュメモリまたはスラッチパッドメモリを備える。また SX-8 までと異なり、SX-ACE では 64 バイトまたは 128 バイト単位で外部メモリへのアクセスが行われるという点で GPU と同じであるとともに、これまでのバンクコンフリクトさえ避ければよかったベクトルプロセッサ向けの FFT アルゴリズムが使用できなくなったことを意味する。そこで 2008 年に提案した GPU 向け 3 次元 FFT アルゴリズムを SX-ACE 向けに拡張する方向で実装・評価を行った。

3D-RISM の計算で用いる 3 次元 FFT は倍精度複素数でサイズが $256 \times 256 \times 256$ または $512 \times 512 \times 512$ を用いる。いずれも 2 のべき乗であるため演算効率がよく、また SX-ACE のベクトル長である 256 の倍数ということで実装もシンプルになる。

GPU 向けの 3 次元 FFT アルゴリズムでは X 軸方向と、Y 軸方向、Z 軸方向の各変換に対して別のアルゴリズムを適用することによって GPU メモリが不得意とするアクセスパターンを避けるものである。データが連続アドレスに格納されている X 軸方向の変換についてはオンチップの shared memory を使用しながら複数のスレッドで 1 つの変換を細粒度並列処理する。一方、Y 軸方向、Z

軸方向の変換に必要なデータはとびとびのアドレスに格納されており、すべてをプロセッサ内に集める際には同じキャッシュライン上にあるデータが全て転送されるためメモリバンド幅を著しく浪費する。これを避けるため、複数のデータセットに対して同時に変換を行う multi-row FFT アルゴリズムを用いることによってメモリから転送されたデータを全て無駄なく使用する方法を採用する。レジスタ数やオンチップ shared memory の容量の問題から、Y 軸方向、Z 軸方向の変換を一回のメモリ読み書きで完了することはできないため、それぞれ複数回（通常 2, 3 回）に分割して変換を行うことになる。その結果、メモリアクセス量も増大するが、アクセスパターンが連続であることなどから実効メモリバンド幅も高く、とびとびのメモリアクセスを行う実装よりもはるかに高速になる。

$256 \times 256 \times 256$ の変換は以下の 5 ステップで実現される。

- (1) Z 方向の 16 点 FFT 第一段階
- (2) X 方向の 256 点 FFT
- (3) Z 方向の 16 点 FFT 第二段階
- (4) Y 方向の 16 点 FFT 第一段階
- (5) Y 方向の 16 点 FFT 第二段階

Y 軸、Z 軸方向に関しては 16 点 FFT 2 回によって 256 点 FFT の計算を完了する。X 方向の変換から始まらないことには意味がある。データは通常 $(256, 256, 256)$ の配列に格納されているが、この場合に SX-ACE ではバンクコンフリクトが多発してしまう。これを避けるために配列にパディングを挿入することで大きく性能が改善する。 (x, y, z) 要素は $[x + (256 + 16) * y + ((256 + 16) * 256 + 16) * z]$ のアドレスに格納されるようにしており、また複素数データの実部と虚部は別の配列に格納している。データの初期配置および最終配置ではパディングがない状態を想定して、最初と最後のステップにパディングあり・なしの配列間のコピーを含めている。

$512 \times 512 \times 512$ の変換は以下の 7 ステップで実現される。

- (1) Z 方向の 8 点 FFT 第一段階
- (2) X 方向の 512 点 FFT
- (3) Z 方向の 8 点 FFT 第二段階
- (4) Z 方向の 8 点 FFT 第三段階
- (5) Y 方向の 8 点 FFT 第一段階
- (6) Y 方向の 8 点 FFT 第二段階
- (7) Y 方向の 8 点 FFT 第三段階

Y 軸方向、Z 軸方向に関しては 8 点 FFT 3 回によって 512 点 FFT の変換を完了する。

それぞれ 8 点 FFT と 16 点 FFT に分割しているが、このサイズの上限はベクトルレジスタの数で決定される。SX-ACE は 8 個のベクトルレジスタと 64 個のベクトルデータレジスタを持っているが、ベクトルレジスタはベクトル演算器の入力となるためデータを保持する用途では使用できない。64 個のベクトルデータレジスタで 32 点の複素 FFT のデータを保持するには十分ではなく、2 のべき乗としては 16 点 FFT が最大となる。

X 方向の実装は GPU と大きく異なる。GPU で 256 点 FFT を計算する場合には 64 スレッドで 4 点 FFT を計算し、オンチップ shared memory を介してスレッド間でデータ交換をしながら計算を完了する。256 点 FFT をベクトル長が 256 であるベクトルプロセッサで計算することになるが、1 個の 256 点 FFT の計算に並列度は 256 確保できない。このため 16 個の 256 点 FFT を同時に計算することによって 256 の並列度を確保している。また SX-ACE の ADB はキャッシュであるためデータ保持に利用できるが、Write-Through 方式であるためメモリ書き込み量は増大する。

512 点 FFT の場合は 8 点 FFT 3 回に分解する。1 個の 512 点 FFT で 64 の並列度が確保できるため、4 個の 512 点 FFT を同時に計算することで 256 の並列度が確保できる。ADB へのアクセスが増加するため、メモリ書き込み量が多くなる。ADB 上でデータ交換を行うためにインデックス計算やアドレス計算も必要となるためベクトル命令数も多い。これらの要因により X 軸方向の変換は GPU の場合ほど高速とはならない。

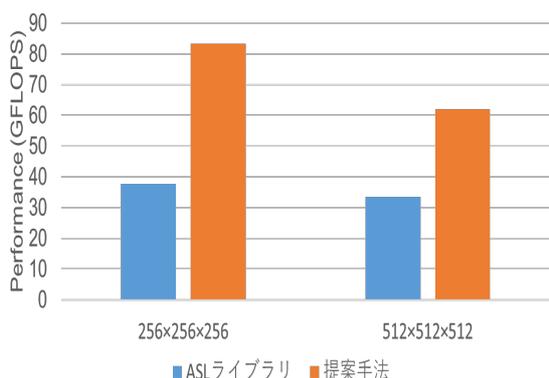
| | 実行時間 | 性能 |
|----------|-----------|----------------------------|
| Z 軸方向(1) | 3.42 msec | 97.8 Gflops 156.6 GB/s |
| X 軸方向 | 9.87 msec | 67.9 Gflops 81.58 GB/s |
| Z 軸方向(2) | 3.32 msec | 100.9 Gflops 161.5 GB/s |
| Y 軸方向(1) | 3.62 msec | 92.5 Gflops 147.9 GB/s |
| Y 軸方向(2) | 3.52 msec | 95.2 Gflops 152.3 GB/s |

上記の表は 256×256×256 の 3 次元 FFT を実行した際の各処理の実行時間および演算性能、メモリバンド幅の計測結果である。Y 軸方向、Z 軸方向では十分なメモリバンド幅が達成されていることがわかる。FFT 計算ではメモリ読み込みと書き込みがほぼ同量であることから STREAM ベンチマーク等よりはバンド幅が低下する。X 方向は Flops、メモリバンド幅ともに低い。Write-Through によるメモリ書き込みも含んだ数値である。また ADB からの Read 時のバンクコンフリクトが回避できていないことと、整数のベクトル命令が多く実行されていることが主な要因である。

| | 実行時間 | 性能 |
|----------|------------|---------------------------|
| Z 軸方向(1) | 32.2 msec | 62.6 Gflops 133.7 GB/s |
| X 軸方向 | 106.9 msec | 56.4 Gflops 80.3 GB/s |
| Z 軸方向(2) | 31.0 msec | 64.8 Gflops 138.3 GB/s |
| Z 軸方向(3) | 31.3 msec | 64.3 Gflops 137.2 GB/s |
| Y 軸方向(1) | 29.9 msec | 67.4 Gflops 143.7 GB/s |
| Y 軸方向(2) | 31.1 msec | 64.7 Gflops 138.1 GB/s |
| Y 軸方向(3) | 29.4 msec | 68.4 Gflops |

| | | |
|--|--|------------|
| | | 146.0 GB/s |
|--|--|------------|

上記の表は 512×512×512 の 3 次元 FFT を実行した際の各処理の実行時間および演算性能、メモリバンド幅の計測結果である。基本的な傾向は 256 の場合と変わらない。データが大きくなったためにバンクコンフリクトを容易なパディングで回避することが難しくなっており、Y 軸方向、Z 軸方向の計算時のメモリバンド幅が低下している。



上記のグラフは我々が開発した実装と SX-ACE を開発した NEC が提供する ASL ライブラリとの性能比較である。ASL ライブラリに関してはパディングパターン、スレッド数を変えて網羅的に性能計測を行い、その中で最高性能となるものを選んでいる。256 と 512 のいずれのデータサイズにおいても ASL ライブラリの約 2 倍以上の性能を達成している。

この 3 次元 FFT を 3D-RISM 及び RMDFT プログラムに取り入れてベクトル化を行なった。テスト系として原子数 1960 のリゾチーム (PDB ID: 6LYZ) の計算を行なった。5.1 の計算手続のうち (3) から (8) の繰り返し部分の計算時間を測定した。計算セルは同じ大きさにして 512×512×512 はグリッド間隔を 256×256×256 の半分とした。

| grid | SX-ACE |
|-------------|---------|
| 256×256×256 | 89.8 s |
| 512×512×512 | 783.2 s |

上記の表は SX-ACE1 ノードでの結果を表す。後に

示す GPU の結果 (5.3 参照) と比べて高速である事がわかる。これは 3D-RISM がベクトルマシンに向いているプログラムである事を示している。また、SX-ACE での計算では、grid 点の増加 (8 倍) に伴い、ほぼ線形 (8.7 倍) に計算時間が増加している。これはプログラムが SX-ACE の性能を十分に引き出している証であると考ええる。

5.3 TSUBAME2.5

GPU クラスターである TSUBAME2.5 において GPU 一つでは 512×512×512 グリッドの大規模計算に必要なメモリを確保できない。そこで最初から MPI を使用した並列プログラムの作成に取り組んだ。また、将来的に大規模並列を実行する事を想定して計算セルを 2 次元分割する事とした。

一般に YX 分割の 3 次元 FFT の正変換を実行する場合は以下のような手順となる。

- (1) X 軸の 1 次元 FFT を実行する。
- (2) Y 軸を揃える通信を行なう。
- (3) Y 軸の 1 次元 FFT を実行する。
- (4) Z 軸を揃える通信を行なう。
- (5) Z 軸の 1 次元 FFT を実行する。
- (6) Z 軸を戻す通信を行なう。
- (7) Y 軸を戻す通信を行なう。

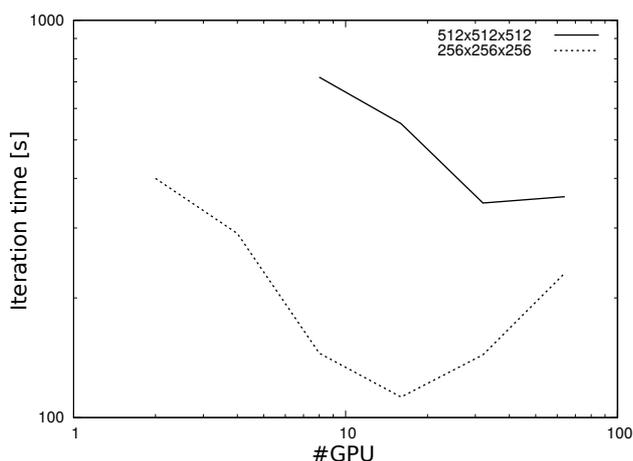
ここで (6) 及び (7) は通信を行なっているだけであり、これを実行しない場合、フーリエ空間での分割が XY 分割となる。つまり実空間とフーリエ空間の分割パターンの差を許容すれば 4 回行われる通信のうち 2 回を行わずに済む。特に大規模ノード数になった場合、FFT の計算時間よりも通信時間の割合が大きくなるため通信を行わない方が計算速度的に有利である。この場合、フーリエ逆変換は

- (1) Z 軸の 1 次元 FFT 逆変換を実行する。
- (2) Y 軸を揃える通信を行なう。
- (3) Y 軸の 1 次元 FFT 逆変換を実行する。
- (4) X 軸を揃える通信を行なう。
- (5) X 軸の 1 次元 FFT 逆変換を実行する。

となる。この実空間とフーリエ空間の分割パターンが違う事についてはプログラムで対応させなければならない。しかし、その対応をするメリット

があると考え、加えて大規模ノードでの実行を念頭においてプログラムの作成を行なった。1 次元 FFT は NVIDIA のライブラリである cufft を使用した。

この 2 次元分割版の GPU 用 MPI プログラムを作成し、TSUBAME2.5 上でテスト計算を行なった。テスト系は 5.2 と同じリゾチームである。



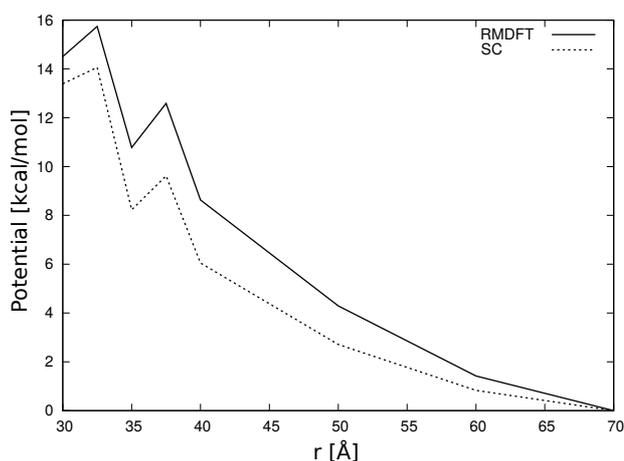
上記の図は、繰り返し部分の計算時間の GPU 数依存性を示す。横軸は GPU 数で、縦軸は計算時間を表す。使用メモリの関係で 256×256×256 では 2GPU から、512×512×512 では 8GPU からとなっている。256×256×256 では 16GPU で 112.6 秒、512×512×512 では 32GPU で 346.8 秒が最も速くなっている。SX-ACE と比較すると 256×256×256 では 16GPU でも及ばなかった。これは FFT における通信の割合が大きくなり、リニアにスケールしないためである。512×512×512 では 32GPU で SX-ACE1 ノードの倍程度の速度を達成している。

[1] A. Nukada, Y. Ogata, T. Endo and S. Matsuoka. In Proceedings of the 2008 ACM/IEEE conference on Supercomputing (SC'08), 5:1—5:11, Austin, IEEE Press, November (2008).

5.4 蛋白質-蛋白質相互作用

溶媒和自由エネルギーの計算精度を調べるために蛋白質-蛋白質相互作用を計算し比較を行なった。低分子は溶媒和自由エネルギーの実験値が存

在するが、蛋白質はその物が難しい。そこで、蛋白質間の相互作用を計測した X 線散乱実験との比較を行なう事で計算精度の検証を行なう事を考えた。対象の蛋白質はリゾチーム (PDB ID: 6LYZ) を用いた。重心間距離に対する相互作用を計算するために 24 個の回転構造を生成し、距離一点について 576 (=24×24) の計算を行い、統計平均を取った。



上記の図は、リゾチーム間の重心間距離と蛋白質間ポテンシャルの関係を示す。実線は RMDFT、点線は SC によるポテンシャルである。RMDFT 及び SC ともに 35Å 付近にローカルミニマムが観察される。これは 34Å 付近にコンタクトがあるという X 線散乱の実験結果と一致する。[2] リゾチームは正に帯電しているため、溶媒の効果がなければ斥力のみとなりコンタクトは発生しない。つまり溶媒の効果がなければリゾチーム同士が接触しない事になる。

一方で計算結果ではローカルミニマムのエネルギーが遠方よりも高いところにありリゾチーム間の距離が離れた方が安定となっている。これは、計算条件と実験条件の違いによるものと考えられる。計算ではリゾチーム 2 個のみの無限希釈系を想定しているが、実際の 10 wt.% の濃度となっている。このリゾチームがより混雑した状況下ではリゾチームがあるリゾチームから離れても別のリゾチームに接近してしまう。そのためある一定以上の距離を取る事ができない。そのような混雑した状態における効果を計算にとり入れる、

もしくは補正を行なう必要がある。

[2] T. Sumi, H. Imamura, T. Morita and K. Nishikawa. *J. Mol. Liquids* 200, 42-46, (2014).

6. 今年度の進捗状況と今後の展望

SX-ACE 及び TSUBAME2.5 上で大規模な生体系における溶媒和自由エネルギーを高速に計算するプログラムを作成した。

今回 SX-ACE を初めて使用したが、様々な性能計測を経てこのプロセッサの特性を大分理解することができた。ベクトル化だけではなく ADB 及びメモリバンクの 2 段階におけるバンクコンフリクト回避が重要であることが判明し、FFT のような規則的なメモリアクセスを行う場合においても工夫が必要である。

今回 SX-ACE で達成された性能は当初想定していた性能の上限と比較するとやや低い。この点は非常に残念であるが、その分マルチノードを使用した並列実行時の強スケーリング性能が期待される。ノード間のデータ転送速度が低いためノード間データ転送がボトルネックとなることは避けられないが、ノード内の計算をどれだけオーバーラップすることができるかがポイントとなる。

大規模計算を実行するにあたりマルチノードマルチ GPU が必須となる TSUBAME2.5 では、3 次元 FFT でノード間通信が発生する。使用ノード数 (GPU 数) が増加するにつれて計算よりも通信の割合が増加する。そのため通信性能が計算速度に大きく影響する。現状の TSUBAME2.5 は InfiniBand が Connect-X2 であるため GPUDirect Version3 が使用できない。そのため GPU から MPI の通信をする時に GPU と InfiniBand 間での直接データ転送 (RDMA) する機能が利用できなかった。この RDMA を利用する事ができれば 3 次元 FFT における通信時間を減少させスケーラビリティを向上させる事が可能と考える。次世代の TSUBAME3.0 等での比較検討を行ないたい。

溶媒和自由エネルギーの計算精度を検証するために行なった計算は、定性的な傾向を示すだけに留まった。これは、実験条件と計算条件の差違に

よる影響が想定よりも大きかった事による。より定量的な議論をするためにはダミーのイオンを分布させるなどして実験条件に近い環境で計算を実行する必要があることがわかった。計算条件を整えて、溶媒和自由エネルギー計算の精度をより定量的に評価するための計算を行なっていきたい。

7. 研究成果リスト

(1) 学術論文

T. Sumi, A. Mitsukake, and Y. Maruyama, *J. Comput. Chem.* 36, 1359-1369, (2015).

(2) 国際会議プロシーディングス

(3) 国際会議発表

(4) 国内会議発表

(5) その他 (特許, プレス発表, 著書等)