

jh160040-NAJ

## 分子動力学計算ソフトウェア MODYLAS のメニーコア およびワイド SIMD アーキテクチャ対応並列化に関する研究

安藤 嘉倫 (名古屋大学 工学研究科計算科学連携教育研究センター)

**概要** 本研究では昨年度に引き続き、汎用分子動力学計算ソフトウェア MODYLAS について次世代のメニーコアおよびワイド SIMD アーキテクチャ (FX100 および Xeon Phi) の性能を發揮させるための並列化チューニングを行った。本年度は SIMD 最適化について、まず問題点抽出のためのコード性能評価を行った。その上で分子動力学計算における 2 つのホットスポット (粒子対での相互作用計算 p2p および高速多重展開法での多極子展開係数から局所展開係数への変換 M2L) でのスレッド並列および SIMD 演算の性能向上を目的として、並列アルゴリズム開発およびコーディングを行った。さらに最適化を進めるにつれ律速段階として顕在化した点電荷分布から多極子への変換 (p2M) と局所展開係数を用いた被相互作用点の電場計算 (L2p) についても SIMD 演算の著しい性能向上を達成した。

### 1. 共同研究に関する情報

#### (1) 共同研究を実施した拠点名

名古屋大学情報基盤センター  
東京大学情報基盤センター

#### (2) 共同研究分野

##### ■ 超大規模数値計算系応用分野

#### (3) 参加研究者の役割分担

全体統括 : 安藤

MPI 並列化性能統括 : 荻野

分子動力学計算のアルゴリズム開発 :

安藤、吉井、藤本、浅野、篠田、岡崎

並列化コーディング (MD 計算全般) :

安藤、藤本、遠藤

プログラム SIMD 性能評価 : 大島、片桐

Xeon Phi 向け並列化コーディング : 大島

自動性能チューニング技術提供 : 片桐

並列化コーディングおよび並列アルゴリズム

開発 (OpenMP, SIMD) : 鈴木

### 2. 研究の目的と意義

研究の目的 現在分子動力学 (MD) 計算は、化学、物理、生物、およびウィルス学といった様々な学問分野において実験とならぶ解析ツールとして広く普及している。加えて工業

分野においても分子の特性を活かしたナノ機能性材料や高分子材料を設計する際に MD 計算により得られる知見が不可欠になりつつある。分子内・分子間の相互作用には様々あるが、分子集団系の物性をシミュレーションにより正しく再現するには長距離性のある静電相互作用を精度よく計算する必要がある。しかしながら、長距離原子間静電相互作用を含めた実用の研究において MD 計算であつかえる原子数および計算時間は、「京」コンピュータといった最新鋭のスーパーコンピュータを用いたとしても 1 千万原子系 (空間サイズとして 50 ナノメートル立方程度) に対する数 100 ナノ秒の計算<sup>[1]</sup>が限界である。より大規模かつ長時間な MD 計算を行うことで、上記学問分野におけるブレークスルーが期待できるだけでなく、より高精度な材料設計が可能になると期待される。

本研究では、昨年度に続き汎用分子動力学計算ソフトウェア MODYLAS<sup>[2]</sup>に対して FX100 および Xeon Phi といった最新のメニーコア・ワイド SIMD 型の計算機、さらには将来のエクサスケールマシンの性能を發揮させるためのスレッド並列およびワイド SIMD への最適化チューニングを行う。必要に応じてハードウェアに適合したアルゴリズム開発

についても行い、将来的にエクサスケールマシンと MODYLAS によって数億から 10 億原子系での  $1\mu$  秒を超えた実用的な MD 計算を目標とする (図 1)。

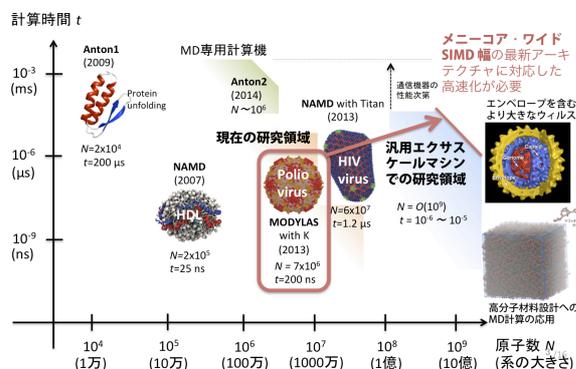


図 1 長距離静電相互作用計算を含む MD 計算で扱える原子数および時間スケールの変遷。

**研究の意義** 分子動力学ソフトウェア MODYLAS は「京」コンピュータの全ノード規模での並列化計算に対応した MPI 並列化チューニングが施されており、昨年度 JHPCN 課題において FX100 の 32 コアでの OpenMP 並列化チューニング、さらには XeonPhi を用いた 240 コアまでのスレッド並列性能検証を進めた<sup>[3]</sup>。一方スーパーコンピュータの進化はノード数の増加が頭打ちになり、ノード当たりのコア数および SIMD ベクトル長 (SIMD 幅) を増加させシステム全体の演算性能を向上させる方向にある。SIMD 幅は「京」において 128bit であったのに対し、FX100 では 256bit、最新の XeonPhi では 512bit ある。さらにポスト「京」では ARM アーキテクチャを用いた 512bit SIMD 幅になる予定であることが既に発表されている。これら次世代のマシンでは、コア数および SIMD 幅の増加によって現行のプログラムの各所でスレッド並列性および SIMD 並列性が確保できなくなる。本研究でのメニーコアおよびワイド SIMD アーキテクチャ向け並列性確保のためのチューニング、計算負荷の均等化による並列化効率の向上、および階層キ

ャッシュの最適化についての研究は、次世代マシン上で高効率な並列化 MD 計算を実現するに不可欠である。

### 3. 当拠点公募型共同研究として実施した意義

メニーコア・ワイド SIMD といったハードウェアの日進月歩の進化に MD 計算ソフトウェアが対応していくためには、最新のアーキテクチャの基本性能および対応コーディングに詳しいコンピュータ・サイエンスの研究者と MD シミュレーションを用い実際の研究を実施している者とが協力した学際領域分野の研究が不可欠である。本研究では両分野の専門家が共同研究者として参画している。

本課題に参加するコンピュータ・サイエンスの研究者はハイパフォーマンスコンピューティング全般を専門とし、特に自動性能チューニング(AT)に詳しい研究者 (片桐)、最新のメニーコア技術に詳しい研究者(大島)、並列化技術全般に詳しい研究者 (鈴木、坂下)、および MPI 並列化技術に詳しい研究者 (荻野) から構成される。複合・階層的な最新のメニーコア・ワイド SIMD 型の計算機システムを使いこなす上で不可欠な人員構成である。一方、MD シミュレーションの研究者は、MD 計算ソフトウェア MODYLAS の基本設計に最初から関与しかつ MD 計算の基本原理に詳しい研究者 (安藤、藤本、吉井、岡崎) および現行の MODYLAS 並列化の内容に詳しい研究者 (安藤、藤本、浅野、坂下) および粗視化 MD 計算に詳しい研究者 (篠田) から構成される。両者の協業により、プログラミングレベルでの並列性能チューニングだけでなく、必要に応じてハードウェアに適した MD 計算のアルゴリズム開発・実装による性能向上も期待される。

### 4. 前年度までに得られた研究成果の概要

MODYLAS では長距離静電相互作用計算

に高速多重極展開法(FMM, 図 2)を用いている。まず FX100 において MD 計算におけるホットスポット二箇所,

- ・Lennard-Jones 相互作用および静電相互作用の粒子対計算 p2p
- ・高速多重極展開法(FMM)での多極子から局所展開係数への変換 M2L

についてのメニーコア対応スレッド並列最適化を行い顕著な性能向上を得た (図 3)。詳細は昨年度の JHPCN 最終報告書<sup>[3]</sup>に譲る。概要として、オリジナルコードではスレッド並列対象 do ループ長が 40 程度しかなく、32 スレッド実行時 50%近い負荷不均衡が生じていた。該当 do ループ長を伸張 (改良その 1) さらにスレッド並列化位置の変更 (改良その 2) により、スレッド間不均衡 1%以下と劇的に縮小され、かつスレッドあたりの計算粒度が約スレッド数倍になった効果などによりスレッド単体性能についても向上した。FX100 で最適化したコードを XeonPhi に移植した上での測定において 120 スレッドまで負荷均衡化および高速化を達成した(図 4)。M2L について、オリジナルコードでは最下層および二階層目については均一なスレッド並列性が確保される一方、より上層においてはチャンクサイズの不備によりインバランスが生じていた。上層部チャンクサイズの最適化を行った結果、3 階層目以降においても均一な並列性を確保できるようになり高速化が達成された。

課題として p2p, M2L のさらなるスレッド並列最適化, XeonPhi 120 スレッド以上でのスケールアップ, プログラム全般での不要なスレッド間バリアの削減, p2M, L2p で使用しているユーザー定義関数 algndr (ルジャンドル倍多項式) での整数演算の効率化, および if 文の削減などを挙げた<sup>[3]</sup>。さらにメニーコアへの対応とともに次世代スーパーコンピュータの性能を引き出すために不可欠なワイド SIMD 幅を活用できるような性能チューニングを今年度 JHPCN 課題として提案した。

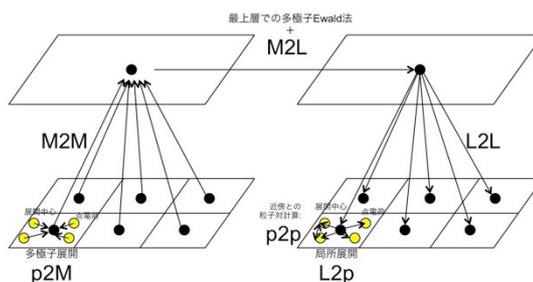


図 2 高速多重極展開法 (FMM)の主要演算. p2p および M2L のほか, 点電荷から多極子展開係数への変換 p2M, 多極子のマージ M2M, 局所展開中心の移動 L2L, および局所展開係数をもちいた点電荷上の電場の計算 L2p から構成される。

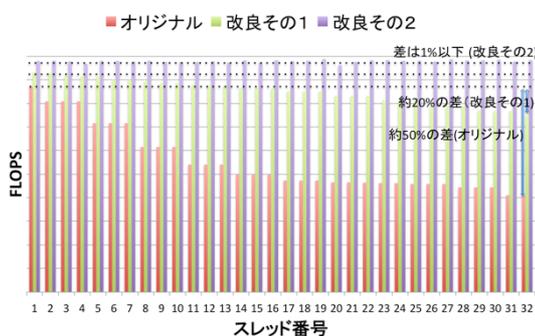


図 3 FX100 において p2p を 32 スレッド実行した時のスレッド番号別演算性能。

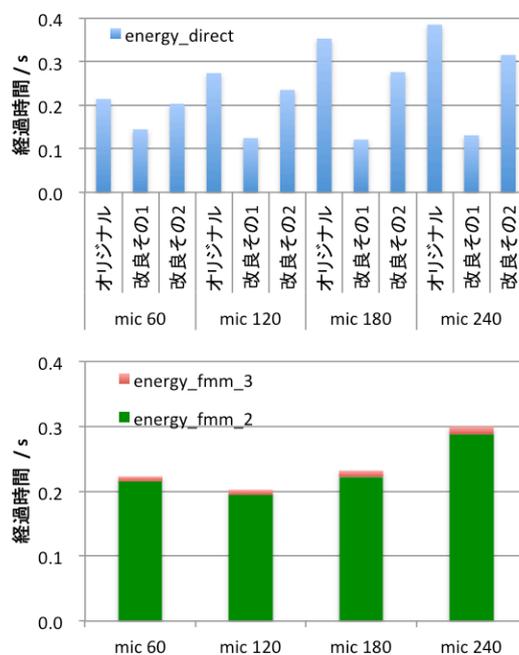
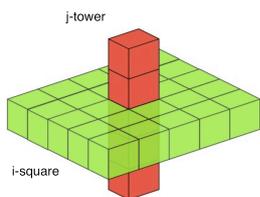


図 4 Xeon Phi での p2p (上), M2L+L2L(下)の測定値. 後者については階層ごとの合計値. スレッド数 60, 120, 180 および 240.

### 5. 今年度の研究成果の詳細

まず p2p について、さらに 2 つのスレッド並列アルゴリズムを提案した。図 5(上)では粒子対計算における自原子  $i$  が所属するサブセルを  $j$ -tower に直交する方向にまとめた平面  $i$ -square ごとにスレッドに割り振る方法で、スレッドあたりの粒度が  $25 \times 5$  セル対と大きく演算性能向上が期待される(改良その 3)。図 5(下)は、昨年度提案した改良その 1 の拡張であり  $i$ -square に直交する  $i$ -column 単位でスレッドに割り当て粒度を高めている(改良その 4)。いずれの方法においても `nowait` 指示句を用いることで有効 `do-loop` 長を伸ばすとともに不要なスレッド間バリアを削減している。

#### i-square 単位スレッド並列化

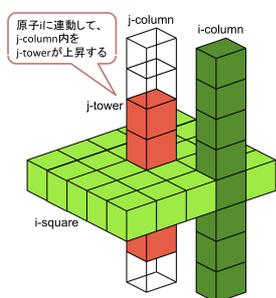


実際には、 $i$ -square 内の有効なセル数は  $(j_{xb}, j_{yb})$  の位置により 1~25 と変化する

```

!$omp parallel
!! j-tower中心(jxb, jyb, jzb)の移動
!! 運動してi-square中心(jxb, jyb, iz)も移動
do 水平面内(jxb, jyb)の移動
!$omp do schedule(dynamic,1)
do 鉛直方向 jzbの移動
j-tower内の相互作用パラメータのバック
do i-square内をセル(ix, iy, iz)が移動
do セル(ix, iy, iz)内の原子i
do j-tower中の原子j
ペア(原子i, 原子j)について
物理量を計算
end do
end do
end do
!$omp end do nowait
end do
!$end parallel
    
```

#### カラム単位スレッド並列化



$i$ -column全体を1スレッドが担当 ( $i_z$ 方向セル数 $\times j$ セル対内の原子ペアを計算)

```

!$omp parallel
do j-towerの水平方向(jxb, jyb)の移動
運動してi-square中心(jxb, jyb)も移動
j-column内の相互作用パラメータのバック
!$omp do
do i-columnのi-square内水平方向移動
i-column内原子→j-tower鉛直方向位置(jzb)
テーブルを作成
do i-column内の原子i
テーブルを参照してj-tower位置を特定
do j-tower中の原子j
ペア(原子i, 原子j)について
物理量を計算
end do
end do
end do
!$omp end do nowait
end do
!$end parallel
    
```

図 5 新たに提案した p2p のスレッド並列化アルゴリズム。自原子の所属する平面  $i$ -square 単位のスレッド並列化(上), および自原子の所属する柱  $i$ -column 単位でのスレッド並列化(下)。

$N \times N \times N$ : 計算領域セル数,  $N_i$ : スレッド数  
セル内の原子数を40として評価

	粒度 (計算原子対数)	スレッド数 $N_i$ の上限			
		$N \times N \times N$	$2 \times 2 \times 2$	$4 \times 4 \times 4$	$8 \times 8 \times 8$
オリジナル	cell内分割 $8000/N_i$	40	40	40	40
改良その1	column内分割 $8000N/N_i$	$40N$	80	160	320
その2	cell単位 8000	$25N^3$	200	1600	12800
その3	square単位 $8000^3 \sim 200000$	$N(N+4)(N+4)$	72	256	1152
その4	column単位 8000N	$25N^2$	100	400	1600

\* MPI領域境界では $i$ -squareがクリッピングされ、有効なセル数が25から最低1まで減少する

表 1 p2p スレッド並列における粒度およびスレッド数上限。  $N$  は軸当たり担当サブセル数。

#### 16コア実行 (1CMGに1MPIプロセス)

	秒	GFLOPS	ピーク比	L1Dミス率	L2ミス率	SIMD化率
オリジナル	6.20	130.4	23.1%	0.40%	0.00%	96.56%
改良その1	4.83	168.2	29.9%	0.14%	0.00%	96.56%
その2	4.23	192.2	34.1%	0.25%	0.00%	96.56%
その3	4.02	200.9	35.7%	0.20%	0.00%	96.56%
その4	4.11	197.0	35.0%	0.08%	0.00%	96.56%

#### 32コア実行 (1ノードに1MPIプロセス)

	秒	GFLOPS	ピーク比	L1Dミス率	L2ミス率	SIMD化率
オリジナル	4.74	170.4	15.1%	0.45%	0.02%	96.56%
改良その1	3.14	257.1	22.8%	0.30%	0.02%	96.55%
その2	2.87	284.9	25.3%	0.22%	0.04%	96.56%
その3	2.21	365.9	32.5%	0.09%	0.01%	96.56%
その4	2.32	349.8	31.1%	0.12%	0.02%	96.56%

GFLOPS値はMPIプロセスあたり、SIMD化率= SIMD浮動小数点演算命令率

表 2 FX100 での p2p 性能比較。オリジナルおよび改良コードでの 16 および 32 スレッド実行。

表 1 には提案した方法の粒度およびスレッド数上限  $N_i$  をまとめた。ここで  $N$  は基本セルを MPI で領域分割した際に担当する軸当たりのサブセル数である。オリジナルコードでは粒度が小さく、また  $N$  より  $N_i$  がサブセル内原子数 (~40) であるのに対し、改良その 3 では粒度の拡大, 改良その 4 では粒度の拡大とともに  $N_i$  の拡張に成功している。表 2 には FX100 において p2p を 16 および 32 スレッド実行した際の性能測定結果を示した。16 および 32 スレッドともに改良その 3 のコードの性能向上率が最も高く 35% (16 スレッド), 47%(32 スレッド)の高速化を達成した。これらの成果を論文としてまとめており、近日中に英文誌へ投稿予定である。

一方 p2p の SIMD 化率 (SIMD 浮動小数点演算命令率) については、オリジナルコードは「京」の 128bit SIMD 幅を前提にコーディングされていたが、FX100 の 256bit SIMD 幅

であっても引き続き 96%の非常に高い値を示した。これは元々のコードで最内の j-tower 中の原子 j についてのループ長が 200 程度確保されていたためと考えられる。最内ループの基本的構造は同じのため、改良コードその 1~4 についても同様の SIMD 化率が保たれている。

表 3 p2p 以外の主要演算部の SIMD 化率 (中間報告時点).

	浮動小数点演算ピーク比	SIMD 浮動小数点演算命令率	SIMD 整数演算命令率
p2M	0.7%	0.6%	0.0%
M2L+L2L	12.9%	85.5%	0.0%
L2p	1.0%	15.1%	0.0%
dihedral	1.0%	0.3%	1.7%
SHAKE	0.5%	6.3%	0.7%
RATTLE	0.4%	4.8%	0.5%

表 4 SIMD 最適化後の SIMD 化率.

	浮動小数点演算ピーク比	SIMD 浮動小数点演算命令率	SIMD 整数演算命令率
p2M	2.9%	49.2%	1.4%
L2p	5.4%	55.0%	4.7%
dihedral	4.7%	68.5%	15.3%

一方、表 3 には中間報告時点での p2p 以外の演算部での SIMD 化率をのせた。ホットスポットの 1 つである M2L+L2L については SIMD 浮動小数点演算命令率が 85.5%と高いのに対し、p2M および L2p は 0.6%および 15.1%とあまり SIMD 化されていなかった。その理由は図 6 にあるように、p2M および L2p にはデカルト座標から極座標へ変換する組み込み関数 `cart2angle`、およびルジャンドル倍多項式の組み込み関数 `algnldr` を用いた演算が含まれ、これら組み込み関数内の多重 do ループ内に多くの IF 分岐があるだけでなく SIMD に対するループ長不足が発生していたためである。さらに二面角相互作用の計算 (dihedral) および原子間拘束力の計算

(SHAKE/RATTLE) ルーチンについても

```
do ii=1,ixdiv*lydiv*lzdiv      プロセス所持サブセル数
!$omp do
  do i0=tag(icz0,icy0,icx0),
    & tag(icz0,icy0,icx0)+na_per_cell(icz0,icy0,icx0)-1  サブセル内原子数
    call cart2angle(xta,yta,zta,rad,the,csthe,phi)
    do j0=0,nmax      [典型的にはnmax=4]
    do k0=-j0,j0
      球面調和関数演算
      zphi=dcmplx(0,d0,k*phi)
      g=f*pre(j,iabs(k))*algnldr(j,iabs(k),csthe)  ・SIMDループ長不足
      force, potential 関数参照pre, algnldr
      if分岐
    enddo ! k0
  enddo ! j0
enddo ! i0
!$omp end do nowait
enddo ! ii
```

$$Y_\ell^m(\theta, \phi) = (-1)^{(m+|m|)/2} \sqrt{\frac{2\ell+1}{4\pi} \frac{(\ell-|m|)!}{(\ell+|m|)!}} P_\ell^m(\cos \theta) e^{im\phi}$$

図 6 オリジナルコードでの p2M および L2p の構造。

```
!$omp do
do ii=1,ixdiv*lydiv*lzdiv
  do i0=tag(icz0,icy0,icx0),
    & tag(icz0,icy0,icx0)+na_per_cell(icz0,icy0,icx0)-1
    call cart2angle(xta,yta,zta,rad,the,csthe,phi)
    → rad,the,csthe,phi 配列
  enddo
  do j0=0,nmax      [典型的にはnmax=4]
  do k0=-j0,j0
    do i0=tag(icz0,icy0,icx0),
      & tag(icz0,icy0,icx0)+na_per_cell(icz0,icy0,icx0)-1
      rad,the,csthe,phi 配列に対する球面調和関数演算
      force, potential
      → pre, algnldr をテーブル化
    enddo ! k0
  enddo ! j0
enddo ! i0
enddo ! ii
!$omp end do nowait
```

図 7 p2M および M2L の SIMD 最適化案。

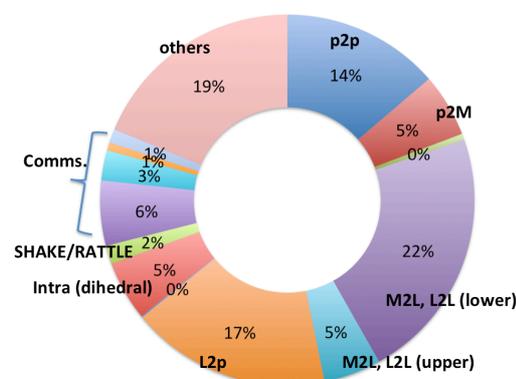


図 8 典型的な入力での計算時間構成 (中間報告時)。NPT アンサンブルでの実行。

SIMD 化率が低く改善の余地があった。二面角相互作用の計算においては IF 分岐分岐の存在が、原子間拘束力の計算では該当 do ループ長の短さが SIMD 化を阻んでいた。

図 8 にあるように、昨年度および今年度 JHPCN 課題を通して行ってきた p2p および M2L のスレッド並列および SIMD 最適化の

結果、中間報告時点で MD の 1 ステップあたりの計算時間にホットスポット以外 (p2M, L2p, dihedral, および SHAKE/RATTLE) が占める割合が 30% 近くに達していた。より一層の高速化にはこれらルーチンに要する計算時間の削減が必須になった。

今年度の後半において、p2M および L2p に対して図 7 に示す案に沿った SIMD 最適化を行った。その過程で、p2M と L2p にあった重複箇所の一重化、不必要な IF 分岐の削除、プリファクターの分離 (ループ前での事前評価) などを行った。その結果 p2M について SIMD 浮動小数点演算命令率が 0.6% であったところ 49.2% へと向上、L2p について 15.1% であったところ 55.0% へと向上させることができた。二面角相互作用計算についても  $\cos$  関数の引数値に対する IF 分岐を削除するなどして 0.3% から 68.5% への向上を達成した。結果として、全体の計算時間に占める p2M, L2p および二面角相互作用計算の割合が最適化前においてそれぞれ 5%, 17%, および 5% であったものを、2%, 3% および 2% へと大幅に減少させることができ、全体の計算時間を 25% 程度削減することに成功した。

これら最適化の結果、さらに著しく律速段階となった M2L についてはスレッド並列の最適化、SIMD 最適化だけではこれ以上の性能向上は難しく、現在アルゴリズムレベルでの見直しを進めている。また図 8 中の others には熱浴や圧力浴の計算が含まれ、これも計算機科学的側面からの速度向上が難しいため、MD 計算方法論からの改良を検討している。演算時間の減少にともない通信 (Comms.) 時間の占める割合も増加しており、新規通信アルゴリズムの実装、およびレーテンシー削減のために、より下層レベルの通信ルーチン (RDMA) の実装も検討している。

## 6. 今年度の進捗状況と今後の展望

本年度は 4 期に分け研究を進める計画で

あった。第一期 (4-6 月) および第二期 (7-9 月) において、昨年度に引き続きメニーコア対応コード最適化を行いつつ、SIMD 基礎評価を FX100 で行った。ホットスポットの p2p について図 5 にある新しいスレッド並列アルゴリズム 2 つの提案および実装を行い、更なる性能向上を得た。SIMD 基礎評価の結果は表 2, 3 のようになり、プログラム全体のより一層の高速化にはこれまで目立たなかった p2M および L2p のスレッド並列および SIMD 最適化が必要であることがわかった。

第三期 (10-12 月) および第四期 (1-3 月) においては、図 7 に提案する p2M および L2p の SIMD 最適化コードを実装し同箇所のスレッド並列最適化にも取り組んだ。その結果、表 4 にあるようにこれら箇所での SIMD 浮動小数点演算命令率を大幅に向上させ、実行時間を削減することができた。さらに dihedral ルーチンについても SIMD 最適化を試み、実行時間を削減することに成功した。結果として、典型的なインプットに対して、全体の計算時間を 25% 程度削減することに成功した。

来年度の JHPCN 課題ではポスト「京」コンピュータと同じ 512bit SIMD 幅をもつ XeonPhi 上での本格的な性能計測を行い、ホットスポットにおける問題点の抽出および解決案を模索していく。ただし XeonPhi では FX100 での詳細プロファイラーのような精密な実行情報を得ることが難しいため、演算性能測定方法の考案から始める必要がある。また本年度の計測からは、ホットスポットにおける SIMD 最適化の程度が FX100 と XeonPhi とで異なることがわかってきており、XeonPhi 上での計算時間の短縮のために、FX100 との汎用性を犠牲にしたある程度の専用のチューニングを施す必要があるかもしれない。

今年度与えられた計算機資源を用いコード改良のための実行テスト、長時間 MD 計算テスト、および実際の研究への応用 (学術論

文欄[i], [ii]) などを行った。FX10 および FX100 とともに期間内に 100%消費した。なお Xeon Phi については研究用のクラスターを用いているため消費率の対象外である。

粒子対計算部 p2p についてのスレッド並列最適化の成果を論文としてまとめており、近日中に英文誌へ投稿予定である。

## 参考文献

- [1] Y. Andoh, et al. *J. Chem. Phys.*, **141**, 165101(2014).
- [2] Y. Andoh, et al., *J. Chem. Theory Comput.*, **9**, 3201 (2013).
- [3] jh150015-NA11 「分子動力学ソフトウェア MODYLAS のメニーコアアーキテクチャ対応並列化に関する研究」安藤嘉倫(課題代表)  
<https://jhpcn-kyoten.itc.u-tokyo.ac.jp/ja/abstract.php?ID=jh150015-NA11>

## 7. 研究成果リスト

### (1) 学術論文

[i] Y. Andoh, N. Yoshii, A. Yamada, S. Okazaki, Evaluation of atomic pressure in the multiple time-step integration algorithm, *J. Comput. Chem.*, **38**, 704 (2017).

[ii] Y. Andoh, S. N. S. Mohamed, S. Kitou, S. Okazaki, Structural ordering of lipid bilayers induced by surfactant molecules with small hydrophilic head group, *Mol. Siml.*, in press.

### (2) 国際会議プロシーディングス

該当なし

### (3) 会議発表(口頭, ポスター等)

[i] 安藤嘉倫, 鈴木惣一郎, 大島聡史, HPCS2016 オーガナイズドセッション「分子動力学計算ソフトウェア MODYLAS のメニーコアアーキテクチャ対応並列化に関する研究」, HPCS2016, 仙台 (2016). 口頭

[ii] Y. Andoh, N. Yoshii, A. Yamada, S. Okazaki, An attempt to estimate correct atomic pressure in the multiple time-step integration algorithm, ICMS2016, Shanghai (2016). ポスター

[iii] 安藤嘉倫, 吉井範行, 山田篤志, 岡崎進, マルチタイムステップ数値積分法(RESPA)に基づく分子動力学計算での圧力値算出における問題, 第30回分子シミュレーション討論会, 大阪 (2016). 口頭

### (4) その他(特許, プレス発表, 著書等)

該当なし