［大学 ICT 推進協議会 2013年度 年次大会論文集より転載］

東北大学サイバーサイエンスセンターにおける 分子動力学シミュレーションコードの高速化支援について

森谷 友映†, 佐々木 大輔†, 山下 毅†, 小野 敏†, 大泉 健治†, 小松 一彦‡, 江川 隆輔‡, 小林 広明‡
†東北大学　情報部　情報基盤課
‡東北大学サイバーサイエンスセンター　スーパーコンピューティング研究部
t-moriya@isc.tohoku.ac.jp

概要：東北大学サイバーサイエンスセンター (以下，本センター) は，全国共同利用設備としての大規模科学計算システムを最大限に活用するために，計算科学者と本センターの計算機科学者が連携しながら，プログラムの高速化技法の研究・開発に取り組んでいる．本稿では，分子動力学シミュレーションの高度化を例として，本センターの高速化支援の取り組みについて概説する．

1. はじめに

東北大学サイバーサイエンスセンター (以下，本センター) は，全国共同利用設備としての大規模科学計算システムの管理・運用と，本システムを最大限に活用可能なプログラムの高速化技法や新しいシミュレーション技術の研究・開発に取り組んでいる．1997 年から行っている計算科学分野の利用者との共同研究を通じて，さまざまな分野における実アプリケーションの最適化や並列化の高速化支援を行っている．また，センター独自の共同研究に加え，全国の情報基盤センター等と連携して，学際大規模情報基盤共同利用・共同研究拠点 (JHPCN) や革新的ハイパフォーマンス・コンピューティング・インフラ (HPCI) を構成し，共同研究を実施している．以上のように本センターでは，利用者である計算科学者と本センターの計算機科学の専門家が密に連携しながら，科学・工学の恒常的な進歩を支える共同研究，利用者プログラムの高速化支援活動を推進している．

本稿では，これらの取り組みの中でも特に，産業技術総合研究所と本センターとの共同研究として取り組んだ「分子動力学シミュレーションの高度化」について紹介する．本稿の構成は以下の通りである．2 節では，大規模科学計算システムついて概説する．3 節では，本共同研究で対象とする分子動力学シミュレーションと，そのコードの最適化について述べる．4 節で本稿をまとめる．

2. 大規模科学計算システム

本センターでは，大規模な計算機を必要とする全国の利用者の幅広いニーズに応えられるシステム運用を目的に，ベクトル型スーパーコンピュータ SX-9 を提供している．SX-9 は，全 18 ノードでシステムを構成し，1 ノードあたり 1TB の共有メモリと 16 個のベクトルプロセッサを有し，1 プロセッサ（シングルコア）あたりは 102.4Gflop/s の理論演算性能と 256GB/s の高いメモリバンド幅を持つ．特に，ベクトルプロセッサによる高速なベクトル処理能力や，高いメモリバンド幅を必要とする流体解析，気象解析及び電磁界解析の分野によく利用されている．

ベクトルプロセッサは複数のデータを一括して演算を行うベクトル演算が可能であり，配列のそれぞれの要素に演算を行うような計算に適している．このベクトルプロセッサの性能を引き出すためには，プログラム中のベクトル処理可能な部分を示すベクトル化率をできるだけ増やす必要がある．図 1 にプログラムをスカラ処理する場合とベクトル処理する場合の実行時間を示す．

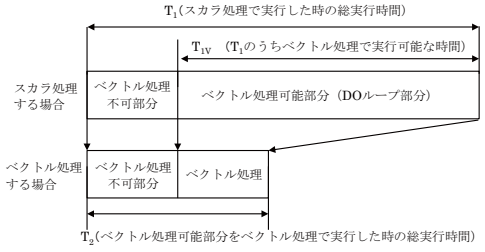

図 1　ベクトル処理による実行時間短縮

スカラ処理する場合の総実行時間を T_1，そのプログラムでベクトル処理が可能な部分の実行時間を T_{1V} とすると，ベクトル化率 α は以下の式で定義される．

$$\alpha \quad = \quad T_{1V} \quad / \quad T_1$$

また，そのプログラムをベクトル処理する場合の性能向上比 P は，スカラ処理性能とベクトル処理性能の比を β とすると以下の式で定義される．

$$P \quad = 1 \quad / \quad (\quad (1-\alpha) \; + \; \alpha \; / \; \beta \quad)$$

この式から，図 2 に示すベクトル化率と性能向上比の関係が導かれる．ベクトル化率が 8 0 %以下では大きな性能向上は見られず，ベクトル化率が 90 %を超えると急速に性能が向上する．そのため，ベクトル型スーパーコンピュータで高い実行性能を得るためには，ベクトル化率を 100 %にできる限り近付ける必要がある．

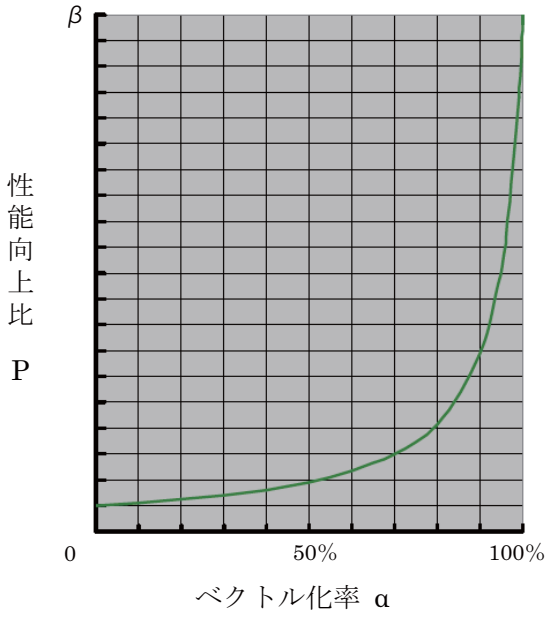

図 2　ベクトル化率と性能向上比

また，大規模なデータをベクトル演算器で効率的に処理するためには，メモリから演算器へのデータ供給を行う必要がある．SX-9 はインターリーブ構成のメモリを持ち，複数のメモリバンクに並列に読み書きを行い高速なデータ転送を実現する．メモリバンクは 32,768 個に分割され，複数のメモリバンクにまたがって連続アクセスする場合が最も効率的で高速なデータ転送となる．逆に不連続でデータアクセスした場合，同一メモリバンクに対して同時にアクセスが発生することがある．これをバンクコンフリクトと呼ぶ．SX-9 におけるバンクコンフリクトは，CPU ポート競合とメモリネットワーク競合の 2 つに分類され，CPU ポート競合は CPU 内における同一のポートにロード・ストアが集中した時に発生する．メモリネットワーク競合は同一のメモリバンクへのアクセスで発生する競合や，CPU と主記憶装置間の経路上で発生する．バンクコンフリクトが発生するとデータ供給能力が低下し，高速化の妨げの要因になる．これを回避するためには，メモリアクセスの間隔が 2 のべき乗になるのを避けることや，ループのアンロールを行うなどしてメモリアクセス回数自体を減らし，バンクコンフリクトを抑制することが必要となる [1].

3.　分子動力学シミュレーションの最適化

本センターでは，図 3 のようにセンター教員・技術職員・計算機ベンダの技術者が密に連携し，利用者を支援している．本稿では，共同研究に参画する産業技術総合研究所，計算機ベンダ，サイバーサイエンスセンターの三機関で連携して高速化に取り組んでいる，第一原理計算を用いた 3 つの分子動力学シミュレーション（FPSEID，FEMTECK，QMAS）における SX-9 を対象とした最適化について述べる．

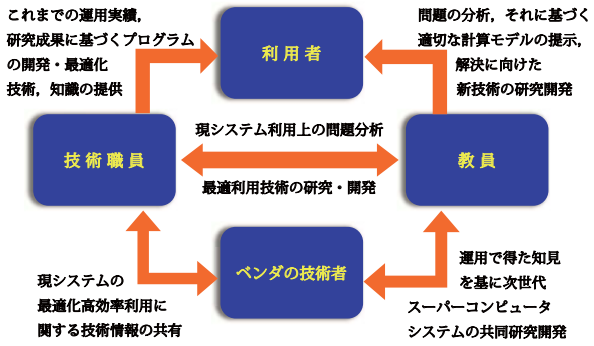

図 3　高速化支援体制

一般的に，最適化を行うためには，まずプログラムの特性を把握する必要がある．SX-9 が標準で備える簡易性能解析ツールを用いて，プログラムの解析結果を取得しどのような最適化が有効かを検討し，最適化を施す．以下に，本報告で対象とする分子動力学シミュレーションについて，それぞれコードの概要，解析結果，最適化とその効果を述べる．

3.1 FPSEID (First-Principles Simulation tool for Electron-Ion Dynamics)

3.1.1 FPSEID の概要

FPSEID は，時間依存密度汎関数理論に基づいて，電子励起が引き起こす物質中の過渡的なダイナミクスを数百フェムト秒の時間スケールで計算することを目的に作成されたプログラムであり，時間依存シュレディンガー方程式を数値計算する際に，高精度な近似公式 (鈴木-Trotter 公式) を利用し通信頻度を減らした並列計算を実現している．これまでの成果としては，光励起化学反応，高強度レーザー光による物質構造変化，高速イオンと物質の衝突に付随する諸現象のシミュレーションを行っており，フェムト秒レーザーやヘリウムイオン顕微鏡などを用いた先進的な実験技術による研究の推進に貢献することが期待されている [2] [3] [4].

3.1.2 ベクトル化率の向上

はじめにプログラムの解析を行い，実行時間の長いサブルーチンの選定を行った．今回対象としたのは 3 個のサブルーチンで，それぞれサブルーチン 1，2，3 とする．各サブルーチンの詳細な解析に取り組み，その後，各サブルーチンの高速化に取り組む．

サブルーチン 1 における解析結果を表 1 に示す．これを見るとベクトル化率が 49.66 %となっており，ベクトルプロセッサへの移植に際して性能向上の可能性があることが分かる．また，図 4 に示すループを見ると，データの格納位置を保持する i2g 配列をインデックスとして他の配列を参照するリストベクトルが用いられている．リストベクトルにより，ベクトル演算が可能かの判断が難しくなる．図 5 では作業配列に計算結果を一度格納し，作業配列の値を参照することでリストベクトルを解消しベクトル化を促進することができる．作業配列を用いてリストベクトルを解消することで，ベクトル化率が 49.66 %から 99.06 %と向上し，対象とするサブルーチンの実効性能は 16.7 倍に向上した．

表 1　サブルーチンの解析結果

PROC. NAME	EXCLUSIVE TIME[sec]	%	MFLOPS	V.OP RATIO	AVER. V.LEN	BANK CONFLICT CPU PORT	BANK CONFLICT NETWORK
orignal	75.553	6.2	29.6	49.66	255.9	0.176	0.277
modified	4.520	0.4	494.8	99.06	255.9	0.514	3.489

+: 最適化されなかったループ
|: ループ構造の範囲
V: ベクトル化されたループ・配列式

```
+------>  do ig=1,nxyz
|            jg=i2g(ig)
|            rhog(jg)=rhog(jg)*fdump(ig)
+------    enddo
```

図 4　オリジナルコード

```
          icnt=1
V------>   do ig=1,nxyz
|            rhog_tmp(icnt)=rhog(i2g(ig))
|            fdump_tmp(icnt)=fdump(ig)
|            icnt=icnt+1
V------    enddo

V------>   do ig=1,nxyz
|            rhog_tmp(IG)=rhog_tmp(IG)*fdump_tmp(IG)
V------    enddo

          icnt=1
V------>   do ig=1,nxyz
|            rhog(i2g(IG))=rhog_tmp(icnt)
|            icnt=icnt+1
V------    enddo
```

図 5　ベクトル最適化コード

3.1.3 バンクコンフリクトの削減

サブルーチン 2 における解析結果を表 2 に示す．全体の実行時間に対してバンクコンフリクトの占める割合が大きいことが分かる．該当のサブルーチンでは複素数型配列を利用しているために飛びアクセスが発生していると考えられる．このため，複素数を実数部と虚数部に分け実数型配列に保存し，メモリ上の連続する領域へのアクセスとすることにより，速度低下やバンクコンフリクトを招く要因となりうる飛びアクセスの削減が期待できる．図 6 に最適化前の複素数型配列を利用したオリジナルコード，図 7 に最適化後の複素数を実数部と虚数部に分け実数型配列にしたベクトル最適化コードを示す．この最適化の結果，サブルーチン単体の実行時間は 1,453 秒から 996 秒となり 457 秒短縮することができた．

表 2　サブルーチンの解析結果

PROC. NAME	EXCLUSIVE TIME[sec]	%	MFLOPS	V.OP RATIO	AVER. V.LEN	BANK CONFLICT CPU PORT	BANK CONFLICT NETWORK
orignal	1453.723	41.5	13481.2	99.62	255.9	255.084	821.097
modified	996.058	34.9	19675.5	99.63	255.9	234.856	466.970

```
V----->          DO 101 JG=1,NXYZ
V-----      101   RHO2(JG)=(0.D0,0.D0)
       *VDIR NODEP(RHO1)
V----->          DO 100 IG=1,NXYZ
|               JG=J2G(IG)
V-----      100   RHO1(JG)=P(IG)
       中略
          CALL FFT3BX_ASL(NRX,NRY,NRZ,NXYZ,RHO1,RHO2,WSAVE_XYZ,IFAC_XYZ)
V----->        do ig=1,nxyz
|              VG(ig)=VGG(ig)+Vloc(ig)
V-----         enddo
V----->          DO 300 I=1,NXYZ
|               fac=dt*dreal( vg(i) )
V-----      300   RHO2(I)=dcmplx( dcos(fac),-dsin(fac) )*RHO1(I)
          CALL FFT3FX_ASL(NRX,NRY,NRZ,NXYZ,RHO2,RHO1,WSAVE_XYZ,IFAC_XYZ)
       *VDIR NODEP(RHO2)
V----->          DO 110 IG=1,NXYZ
|               JG=J2G(IG)
V-----      110   P(IG)=RHO2(JG)
```

図 6　オリジナルコード

```
      *VDIR NODEP
V-----> DO IG=1,NXYZ
|       JG=J2G(IG)
|       RHO1_R(JG)=dble(P(IG))
|       RHO1_I(JG)=aimag(P(IG))
V-----  ENDDO
      中略
        CALL FFT3BX_ASL_NEW(NRX,NRY,NRZ,NXYZ,
       &  RHO1_R,RHO1_I,WSAVE_XYZ,IFAC_XYZ)
V-----> do I=1,nxyz
|       VG_R(I)=VGG(I)+Vloc(I)
|       fac=dt*VG_R(I)
|       RHO2_R(I)=(dcos(fac)*RHO1_R(I))-((-dsin(fac))*RHO1_I(I))
|       RHO2_I(I)=(dcos(fac)*RHO1_I(I))-((dsin(fac))*RHO1_R(I))
V-----  enddo
        CALL FFT3FX_ASL_NEW(NRX,NRY,NRZ,NXYZ,RHO2_R,RHO2_I ,WSAVE_XYZ,IFAC_XYZ)
      *VDIR NODEP
V-----> DO IG=1,NXYZ
|       JG=J2G(IG)
|       P(IG)=dcmplx(RHO2_R(JG),RHO2_I(JG))
|       VG(IG)=dcmplx(VG_R(IG),0.0d0)
V-----  ENDDO
```

図 7　ベクトル最適化コード

3.1.4　ファイル読み込みの最適化

サブルーチン 3 の解析結果を表 3 に示す．これによりサブルーチン 3 ではベクトル化率が 1.2 %となっている．詳細な解析により，ループ中でファイルからデータを読み込んだ直後に，読み込んだデータに対して演算が実行されていることが分かった．そこで，ファイルの読み込みと演算を別々に処理するよう，最適化を施した．その結果，ベクトル化率が 1.2 %から 98.99 %と向上し，サブルーチン単位の実行時間は 219.190 秒から 0.095 秒へ短縮することができた．

表 3　サブルーチンの解析結果

PROC · NAME	EXCLUSIVE TIME[sec]	%	MFLOPS	V.OP RATIO	AVER. V.LEN	BANK CONFLICT CPU PORT	NETWORK
orignal	219.190	12.5	4.1	1.20	248.1	0.395	0.057
modified	0.095	0.0	0.0	98.99	248.0	0.017	0.061

3.1.5　自動最適化レベルの変更

この節では，自動最適化の見直しによる高速化の検討を行った．従来は，規定レベルのベクトル化処理を行う-Cvopt オプションを用いていたため，最大限のベクトル化処理を行う-Chopt オプションを採用した．-Chopt を使うことで最適化及びベクトル化をより積極的に促進するメリットがある反面，命令の並び換えに対する最適化が基本ブロックをまたがって行われるため，命令の移動に伴う副作用が生じる可能性がある．しかし，本プログラムの場合は，この最適化レベルを変更しても，演算結果に影響を与えることなく，実行時間を 22 %短縮することが確認できた．これにより，個々の最適化と併せて，プログラムの振る舞いを見ながら適切なコンパイルオプションを選択することが重要であることがわかる．

3.2　FEMTECK(Finite Element Method based Total Energy Calculation Kit)

3.2.1　FEMTECK の概要

FEMTECK は原子レベルのミクロな系の性質を量子力学に基づいてシミュレーションするプログラムであり，アルコール等の分子性液体や燃料電池・リチウム電池の電解質等，非常に複雑な系の第一原理分子動力学計算に適用されている．有限要素法を基底関数として使用しているため，並列計算機での実行に向いており，特にスカラ型並列コンピュータ上ではピーク性能の 30 %以上の実効性能を出すことができる．一方，ベクトル型スーパーコンピュータへの対応は進んでおらず，今後，スカラ型並列コンピュータでは扱うことが困難であるような大規模なシミュレーションをベクトル型スーパーコンピュータで実行することが求められている [5].

3.2.2　ベクトル化の促進

前述の様に，本プログラムはこれまでに実行されてきた計算プラットフォームが異なるため，SX-9 における高効率実行を実現するためには，ベクトル化を促進する方針で最適化作業を行った．オリジナルコードでは，ループ中にサブルーチンコールがあり，これがベクトル化の阻害要因となっていた．このサブルーチンコールをコンパイルオプション (-pi) により，インライン展開を行うことで関数呼び出しが不要になりベクトル化が可能となる．

また，変数へのメモリ参照がリストベクトルとなっているループでは，コンパイラにより依存関係の有無が判断できないため自動ベクトル化が行われない．そこで該当するループに依存関係が無いことを確認し，コンパイラ指示行 (!cdir nodep) によりベクトル化を行う．あわせて，ループ中の stop 文や if 文もベクトル化の阻害要因となるため，デバッグ用途で用いられていた stop 文はコメントアウトし，if 文による条件分岐はループの前で行うことで，該当ループのベクトル化を行う．

次にループの繰り返し回数が短く（～16 程度）実行効率が高くない最内側ループに対しては，コンパイラ指示行 (!cdir expand) によりループ展開を行い，よりループ長の長い外側のループに対してベクトル化を行う．また最内側がベクトル化されているもののそのループ長が 256 に満たないループでは，外側のループをコンパイラ指示行 (!cdir unroll) によりループアンローリングを行う．ループアンローリングとはループ本体の演算数を n 倍にし，繰り返し数を 1/n とすることでロード・ストア回数を削減する高速化技法である [1]．コンパイル指示行をプログラム中に挿入することでコンパイル時に自動的にアンローリングされる．アン

ローリング指示行をプログラム中に挿入し，ループをアンローリングすることで，ループ中の配列のロード・ストア回数を削減する．これらの手法によりベクトル化されたループの実行効率を高めることが出来る．

また，FPSEID の時と同様に，自動最適化オプションを-Cvopt から-Chopt に変更し，最適化およびベクトル化を促進する．一つのサブルーチンでループ中の不変式をループ外へ移動したことにより結果に差異を生じたため，該当のサブルーチンに対しては，不変量の移動を抑制するオプション指示行を適用する．

以上で示した最適化によりベクトル化が促進され，オリジナルコードに対して最適化後のコードは実行時間比で 3.4 倍高速化され，プログラム全体のベクトル化率は 92.23 %から 99.17 %に向上することが出来た．さらに，ベクトル型スーパーコンピュータでのピーク性能は 39 %の実効性能を達成することが出来た．

3.3 QMAS(Quantum MAterials Simulator)

3.3.1 QMAS の概要

QMAS は，平面波基底と Projector Augmented-Wave (PAW) 法を用いて，密度汎関数理論に基づき，物質の電子状態および各種物性値を高精度に計算できるプログラムである．新規な計算技術をいち早く導入し活用するためのプラットフォームでもある．これまでの主たる研究対象としては，有機導体/強誘電体，鉄系超伝導体，遷移金属酸化物などを扱ってきた．また，陽電子状態・消滅パラメータの計算が可能であり，半導体中の格子欠陥の研究に利用している．今後，通信部分を減らし，1CPU での実行性能を高め，高メモリバンド幅を活用できるプログラムが求められている [6].

3.3.2 バンクコンフリクトの削減

表 4 に QMAS のサブルーチンの解析結果を示す．これを見るとこのサブルーチンにおいてはバンクコンフリクトが発生しており，これを削減することで性能改善が期待できるが分かった．

表 4　サブルーチンの解析結果

PROC. NAME	EXCLUSIVE TIME[sec]	%	MFLOPS	V.OP RATIO	AVER. V.LEN	BANK CONFLICT CPU PORT	NETWORK
orignal	203.045	10.0	25237.5	99.72	250.5	15.747	105.658

本プログラムを確認すると，図 8 に示すように該当のサブルーチンでは，共通する演算処理やリストベクトルがあった．共通する演算処理はループの外に括りだすことで演算量を削減することができる．また，図 9 のようにリストベクトルを解消するため，作業配列を導入しバンクコンフリクトを削減する．

また，さらなるバンクコンフリクトを削減するためにループアンローリングを行う．

```
  do ii=1,nbase
    do ip=1,nprj(ika)
     zzz=dcmplx(0.d0,0.d0)
     do i=igbgn,min0(igend,ngk(ik))
      zzz=zzz
+         +prg(i,ip,ika,ik)*vpr(kdidx(i,ik),ina)
+         *dconjg(vpr(kdidx(i,ik),inabs(ii))
+         *upg(i-igbgn+1,iibs(ii),katm(inabs(ii))))
     end do
     prjst(ip,ii,ina)=zzz
    end do
  end do
 end do

do ii=1,nbase
  zzz=dcmplx(0.d0,0.d0)

  do i=igbgn,min0(igend,ngk(ik))
   zzz=zzz
+      +vpr(kdidx(i,ik),inabs(ii))
+      *upg(i-igbgn+1,iibs(ii),katm(inabs(ii)))
+      *dconjg(vpr(kdidx(i,ik),inabs(jj))
+      *upg(i-igbgn+1,iibs(jj),katm(inabs(jj))))

  end do
```

図 8　オリジナルコード

```
do ii=1,nbase
     do i=igbgn,min0(igend,ngk(ik))

      upg_tmp(ii)=
   +   upg(i-igbgn+1,iibs(ii),katm(inabs(ii)))
      dconjg_tmp(ii)=dconjg(vpr(kdidx(i,ik),inabs(ii))
   +                  *upg_tmp(ii))
     enddo
enddo

!cdir noloopchg
!cdir outerunroll=8
     do ii=1,nbase
      do i=igbgn,min0(igend,ngk(ik))
       ztmp2(ii)=ztmp2(ii)
   +      +prg(i,ip,ika,ik)*vpr(kdidx(i,ik),ina)
   +      *dconjg_tmp(ii)
      end do
     enddo
     :
!cdir noloopchg
!cdir outerunroll=8
     do ii=1,nbase
      do i=igbgn,min0(igend,ngk(ik))
       ztmp2(ii)=ztmp2(ii)
   +      +vpr(kdidx(i,ik),inabs(ii))
   +      *upg_tmp(ii)
   +      *dconjg_tmp(jj)
      end do
     enddo
```

図 9　ベクトル最適化コード

最適化後の解析結果を表 5 に示す．この最適化の結果，バンクコンフリクトを大幅に削減することができ，実行時間はオリジナルコードに対しておよそ 1.8 倍に改善した．

表 5　サブルーチンの解析結果

PROC. NAME	EXCLUSIVE TIME[sec]	%	MFLOPS	V.OP RATIO	AVER. V.LEN	BANK CONFLICT CPU PORT	BANK CONFLICT NETWORK
orignal	203.045	10.0	25237.5	99.72	250.5	15.747	105.658
modified	110.504	9.7	23257.0	99.55	249.8	5.257	39.183

4. まとめ

本稿では，分子動力学シミュレーションの高度化のために本センターで取り組んでいるプログラムの高速化技法について述べた．本センターが計算資源として提供しているベクトル型スーパーコンピュータ SX-9 の性能を最大限に引き出すためには，ベクトル演算が可能な箇所をできるだけ増やしベクトル化率を向上させることや，効率的なメモリアクセスを実現することによって高いメモリバンド幅性能を十分に引き出す必要がある．

研究の最前線で利用されている 3 つの分子動力学シミュレーションの性能解析に基づいて，ベクトル化を促進させるためのチューニングや，連続したメモリへのアクセス，バンクコンフリクトの削減や抑制などの最適化を行った．これにより，FPSEID, FEMTECK, QMAS のそれぞれで 1.4 倍, 3.4 倍, 1.8 倍の性能向上を達成することができた．

これらの高速化支援により，従来の計算時間では困難であったシミュレーションを実現でき，実験だけでなく計算科学による分子動力学の解析に貢献することができる [2]．今後の課題として FPSEID は並列性能向上と主記憶容量の削減を行い，大規模・長時間のシミュレーションを現実時間内で実行することが挙げられる．FEMTECK は，次期スーパーコンピュータを見据えた高速化を検討することが挙げられる．QMAS は，通信部分の時間を削減し，高メモリバンド幅を活用できるコードが挙げられる．

謝辞

本活動は，共同研究に参画する産業技術総合研究所の石橋章司博士，土田英二博士，宮本良之博士，日本電気株式会社の稲坂純様，萩原孝様，磯部洋子様，百瀬真太郎様に多大なご協力をいただいた．ここにあらためて感謝の意を表する．

参考文献

[1] 東北大学サイバーサイエンスセンター, 高速化推進研究活動報告 第 5 号」, (2011).
[2] ダイヤモンド電子放出デバイスの高性能化の鍵を理論的に解明, http://www.aist.go.jp/aist_j/press_release/pr2013/pr20130913/pr20130913.html
[3] カーボンナノチューブを試験管にした光化学反応，http://www.aist.go.jp/aist_j/press_release/pr2012/pr20120522/pr20120522.html
[4] フェムト秒レーザーによる酸化グラフェンの非熱的還元を提案, http://www.aist.go.jp/aist_j/press_release/pr2012/pr20120118/pr20120118.html
[5] 土田 英二：「有限要素法に基づく第一原理分子動力学法について」，東京大学情報基盤センターニュース Vol.11，特集号 1 (2009).
[6] What’s QMAS , http://qmas.jp/