

jh250040

超大規模高分子系 MD データの位相幾何解析の並列高速化基盤検討

萩田 克美 (防衛大学校)

概要 高分子系の超大規模な MD データに対して、粒子がつながったトポロジーを効率的に解析する基盤の検討として、並列分散での利用技術を検討した。個々の鎖同士のトポロジー的關係を評価する大量の処理を、分散実行する手法を検討した。特に、環状鎖同士が、過渡的に絡み合った状態である「Tight Threading」を判定するコードを開発した。この解析は、2つの環状鎖のペアごとの計算負荷が不均一で大量の処理をようするため、クラウド型計算基盤 MDX-1 上での大量分散並列処理を検討した。そのために、MDX-1 の API を利用した仮想環境制御ツールも独自に開発・公開した。加えて、LAMMPS の分散出力データへのアクセス性向上に向け、HDF5 の VDS 機能を活用した効率的なデータアクセス方法についても検討した。

1. 共同研究に関する情報

(1) 共同利用・共同研究を実施している拠点名

北海道大学 情報基盤センター

東京大学 情報基盤センター

大阪大学 D3 センター

九州大学 情報基盤研究開発センター

mdx I

中島研吾(東大)	スパコン利用技術の助言
佐藤敏文(北大)	精密合成高分子の助言
磯野拓也(北大)	精密合成高分子の助言

(2) 課題分野

大規模計算科学課題分野

(3) 参加研究者一覧と役割分担

参加研究者	役割
萩田克美(防衛大)	総括、コード開発計算実施・議論
村島隆浩(東北大)	副代表、コード開発、計算実施・議論
藤原進(京都工繊大)	計算実施・議論
阪田直樹(東北大)	コード開発、計算実施・議論
下川航也(お茶大)	純粋数学理論の助言
伊達 進(阪大)	MDX 利用技術の助言

2. 研究の目的と意義

(研究目的)

本研究の目的は、高分子鎖で構成される材料系に関して、トポロジー (位相幾何学) の純粋数学と連携した高度な解析技術の創成とともに、放射光での小角散乱実験との連携した大規模なシミュレーション研究に関する技術検討である。

高分子の鎖レベルの解析として、純粋数学由来の高コストで不均一な解析計算を、鎖のペア全てに大量実施する処理系の整備が必要である。また、延伸破壊するゲルの極小角領域の二次元散乱パターンの予測には超大規模なシステムサイズが必要なため分散 IO を用いた計算技術が望まれる。2025 年からの 2 年間程度で、大容量純粋数学的解析処理の実行技術と分散 IO 技術に関する検討を行う。

(研究の意義)

大容量のデータ処理を効率よく実行することは、高分子物理や高分子材料の研究においても死活問題である。計算力を背景にして、大規模 MD シミュレーションと大容量純粋数学的解析処理により科学的真実を導くことは、論文出版という学術研究や、産業応用での設計評価において、競争力の源泉になる。

MDX 等のクラウド技術をうまく活用したデータ解析基盤（独自の専用コード）を作成することは、他社が実施できない競争力の開発として力点を置く。また、高分子物理学者がクラウド技術として利用する参考事例として整備し、研究会・勉強会等を通じて情報提供することで、波及効果も期待できる。

3. 当拠点の公募型共同研究として実施した意義

データ解析処理の効率化の技術検討として、クラウドやシステムソフトウェアの効率的利用を検討するには、基本事項から最新の技術情報までを適切に得ることが重要である。このような状況から、JHPCN センター教員との密な情報共有と対応検討は、本課題の遂行において、きわめて意義が高い。

4. 前年度までに得られた研究成果の概要

該当なし

5. 今年度の研究成果の詳細

(今年度の研究計画の概要)

今年度の研究は、下記の2つのテーマについて、実施する計画である。

- [研究①] Tight Threading を特定する純粋数学由来の解析方法の確立 ～ MDX 大量分散実行～
- [研究②] LAMMPS での分散 IO 技術の利用最適化 ～効率的な解析 platform とノウハウ形成～

[研究①] Tight Threading を特定する純粋数学

由来の解析方法の確立 ～MDX 大量分散実行～

多数の大きな環状鎖が絡まりあった状況では、環状鎖の運動性の一部に異方性がある場合、Topological glass に代表されるように、系の緩和が遅くなる現状が報告されている。これは、Tight Threading (図 1a) と呼ばれる2つの Ring が絡み合い、過渡的にリンクを形成するためと考えられている。Tight Threading している2つの Ring の両端をもち、結び目を保ったまま配置を換えると、Reef knot (図 1b) になる。「Band 手術」は、複数の結び目 (Ring) の一部を切断し別の箇所へ繋ぎ替えて、結び目とリンクのトポロジーを識別する方法である。(Reef knot の両端を切断し、図 1c のように「Band 手術」して1つの Ring として繋ぎ直すと、Square knot になる。)

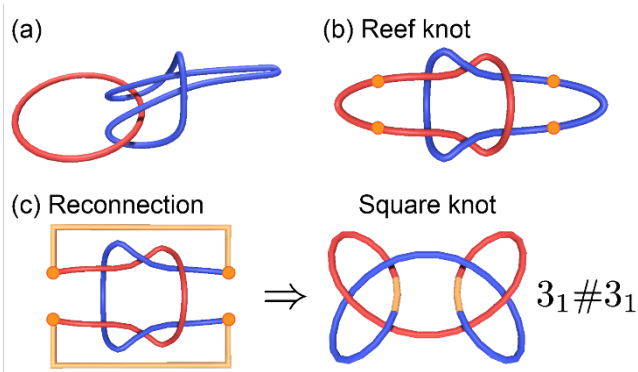


図1 Tight Threading と、Band 手術の模式図

このアイデアを用いた解析法については、最近、Micheletti らが 2024 年に報告していたことが判明した。これを受け、数学的なアイデアの確認のフェーズを終え、当初より予定していた数学的拡張が可能なコードとして構築するフェーズとした。

多数本の環状鎖が周期境界条件中に配置された系において、それぞれの環状鎖が、Tight Threading に関与しているかの確率 p_{TT} を評価したい。環状鎖の本数を M 、Tight Threading に関与している環状鎖の本数を M_{TT} とすると、 $p_{TT} = M_{TT}/M$ である。形式的には、ある環状鎖と残り全ての環状鎖との間での Tight Threading の判定が必要である。また、他の環状鎖は、周期境界条件下でのイメージの配置も考慮する必要がある。な

お、2つの環状鎖の配置が、ある平面により分離可能な組である場合は、**Tight Threading** していないとすぐに判定できることから、計算対象を減らす前処理が可能である。この前処理後に残ったペアに対する **Tight Threading** の判定処理を効率よく、分散並列実施する必要がある。**Tight Threading** の判定処理に必要な時間は、環状鎖の構造に大きく依存することから、一定ではない。

◆Tight Threading の判定処理の説明

上記の通り、前処理によって線形分離が不可能と判定された2つの環状鎖のペアを対象に **Tight Threading** の判定を行なう。

図1に示した **Reconnection** を行う点を決めるために、2つの環状鎖がある状態で、込み入っていない「露出した粒子」を見つける必要がある。具体的には、各環状鎖を構成する各粒子について、その粒子がそれ以外の全粒子（もう一方の環状鎖も含む）から線形分離可能なもの「露出した粒子」を複数リストアップする。2つの環状鎖について、それぞれの「露出した粒子」の距離が最大のものを **Reconnection** の点とする。（ただし、2つの環状鎖がほぼ重なっているような場合は、上記の距離を最大とする条件よりも、露出している向きが重要になる。この対処法の詳細は省略する。）

続いて、**Band** 手術による繋ぎ替え処理を行なう。まず、2つの環状鎖全体の重心を求め、その重心を中心として2つの環状鎖を完全に覆う球を定義する。次に、先に選んだ「露出した粒子」の座標を近傍の2点に分け、端点（計4点）のそれぞれから球面上に半直線を引く。これにより、球面に沿って、互いに交差することなく繋ぎ直すことができ、新たな1つの結び目が形成される。

最後に、上記で得た結び目の不変量を判定し、自明な結び目ではないとき、その2つの環状鎖は「**Tight Threading**」状態にあると決定できる。なお、結び目不変量は **Topoly** や **SnapPy** といった **Python** パッケージで計算できる。

◆MDX 上での分散並列処理対応

Tight Threading 判定の数学的検証として、環を構成する全粒子から等方的に外球と接続させ、トポロジーの統計的解析を実施するプロトタイプコードを作成した。処理が大量であるが並列実施できるので、実装自由度の高い **MDX** を利用して分散並列処理を段階的に実現する検討を行った。

まず、単一の仮想計算機環境（最大 152 コア CPU）内で、サブプロセスを生成し、判定処理をタスク並列で実行するようにした。

次に、複数台の仮想計算機環境を利用した分散処理化への対応を行なった。解析対象範囲のパラメータを実行時オプションで指定可能とし、ssh による遠隔実行に対応させた。これにより、**MDX** 上の制御用の仮想計算機環境から複数の仮想計算機環境を制御し、分散並列処理を実現した。

◆MDX の利用を便利にするツール作成

MDX で複数台の仮想計算機環境を用いて分散実行処理を効率的に行うためには、API を利用した（起動や休止などの）仮想環境操作が望ましい。**MDX** での分散実行処理の実務利用では、通常、ユーザーポータル上の GUI を通じて、仮想計算機環境ごとに、[Action]-[電源]-起動/シャットダウンや、[Action]-[メンテナンス]-[仮想マシンの休止]の操作を行う必要がある。省人化には、GUI を通じた操作ではなく、API 利用を整備する必要がある。特に、「起動」に加えて、リソースを開放するために「仮想マシンの休止」の操作が重要である。

今回はシェルスクリプト利用を念頭にした CLI ツールを独自に整備し、実装は **github** で公開した。（<https://github.com/hypsakata/mdx-tools>）
機能としては、主に、次の操作を実装した。

- 仮想計算機環境の状態のリスト表示
- 「起動」
- 「シャットダウン」
- 「休止」
- API トークンの有効期限の更新

また、その他の API 機能への対応は、**Rest API** での個別カスタマイズを想定した。

なお、MDX での API 利用には、管理者による API 利用の有効化が必要である。MDX のポータルサイトで取得する API 利用の API トークンは、有効期限の更新が必要である。

◆Apache Spark 等での分散処理検討と北海道大学情報基盤センターでの k8s の試行的利用

Apache Spark などの。並列データ処理フレームワークの活用を念頭に、MDX-1 で Apache Spark のスタンドアロンクラスタの動作確認を実施した。これは、後述の利用準備のためである。

次に、北海道大学情報基盤センターで供用されたコンテナクラウド技術 Kubernetes (k8s)について、北海道大学情報基盤センター 棟朝先生と杉木先生の協力を得て、利用可能性を検討した。MDX から、kubectl コマンドで、北大の k8s を操作することも確かめた。円周率のモンテカルロシミュレーションの python コード (pi-spark.py) で、動作することを確かめた。

なお、実解析コードの利用には、現時点で、Topoly python パッケージや Ruby の独自コード等を実行する環境整備が必要である。現状、k8s 上の Pod はメモリが小さく、pip install の実行も難しい。Pod のプリセットの選択 (CPU とメモリの指定) の調整や、必要な環境の軽量化ないし、コードの純 python 化が必要である。この検討は今後課題とした。なお、耐障害性を考慮した分散処理としては、Dask や Ray を用いた軽量なフレームワークも有効な候補であり、この検討も今後の課題である。

◆大容量処理テストのための効率的なデータ作成

現実の材料系では、単純な輪っか (環状鎖) のみならず、それらを接続した構造の多環状体を作成することができる。北大 佐藤先生、磯野先生が合成で作成した多環状体について、特徴的な物性が観察される系について、形状に関する詳細情報などの反映方法を検討した。これらを元に、Tight Threading 解析手法を用いた特徴付けで、複雑なトポロジーが分子鎖同士の貫通状態やダイ

ナミクスに与える影響を定量的に評価することを進めている。

まず、本研究では当初、 $N=100, 150, 200, 300$ の環状鎖数百個を周期境界条件下に配置した系の粗視化 MD 計算を、さまざまな運動異方性の条件で実行して、Tight Threading の統計的状态が異なる「ベンチマーク評価用のデータ」を系統的に取得することを計画した。既存研究との比較のために、 $k=1.5$ の Angle ポテンシャルを課した場合を中心に検討する。既存研究で取り扱われている active polymer の条件は環状鎖の一部分(1/8 領域)の温度を高く ($T=3.0$) 設定することでゲル (固体) の挙動を示す。そこで我々は高温領域の広さや温度の影響について検討した。図 2 に active polymer のスナップショットを、図 3 に active polymer と normal polymer の緩和弾性率の鎖長依存性を示す。鎖長が長いほど緩和が遅くなる振る舞いを示すことは両者に共通する。しかしながら、active polymer においては、実施したシミュレーション時間の範囲内では緩和しない、プラトーを示している。このプラトーはゴムやガラスなどの固体で見られる振る舞いと対応しており、active polymer が Tight Threading によって連結したネットワーク構造を形成することで、固体化していることを示唆している。プラトーを示す固体状態とプラトーを示さない液体状態の違いは、運動性の高い高温領域の広さと温度という 2 つのパラメータに依存することが明らかとなった。

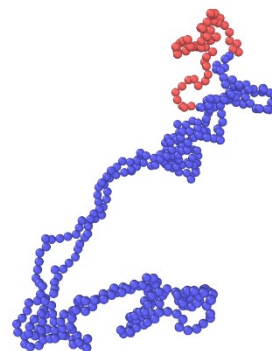


図 2 Active Polymer のスナップショット (赤い：高温粒子、青：低温粒子)。

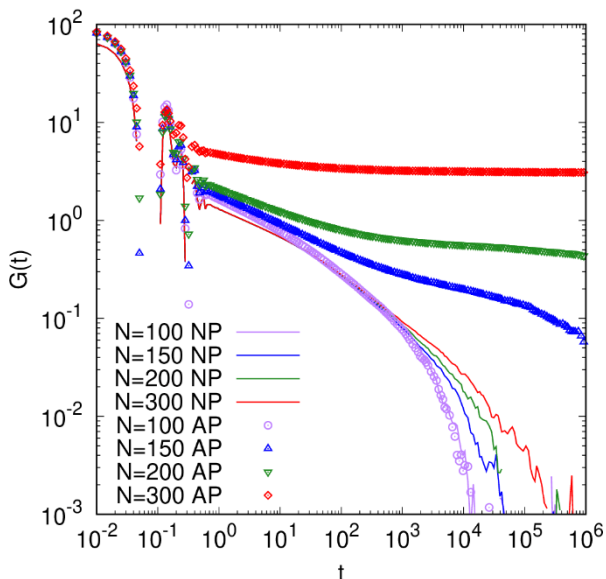


図 3 Active Polymer (AP) と Normal Polymer (NP) に関する緩和弾性率の鎖長依存性

高温領域の広さに関しては、高温部分の温度 $T=3.0$ 、鎖長 $N=200$ の時、 $1/12$ の場合にはゲル（固体）、 $1/20$ の場合にはゾル（流体）となることがわかった。温度の影響については、 $T=2.0$ でゾル（液体）となることがわかり、固体液体転移点は $T=3.0$ と $T=2.0$ の間に存在することがわかった（図 4）。ゲル化を引き起こすためには一部分の運動性を変えることが鍵である。今後は質量の変更や分岐構造の導入といった条件下での検討を進めていく。

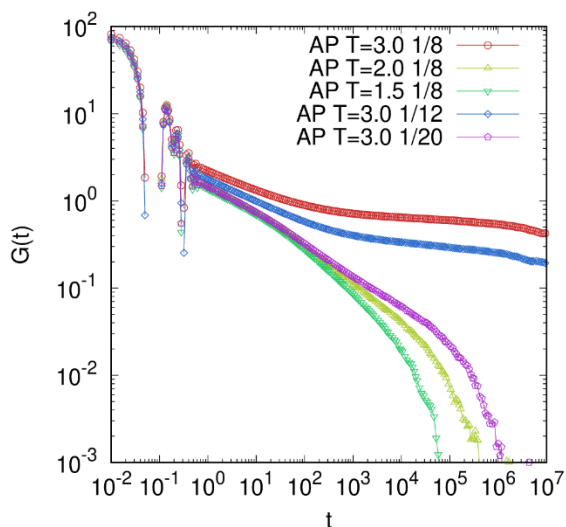


図 4 Active Ring Polymer (AP) の緩和弾性率の高温領域の温度と範囲の依存性

◆レオロジー特性解明の計算手法高速化の検討

高分子材料のレオロジー特性は、緩和弾性率 $G(t)$ の時間依存性に現れる。線形応答理論によれば、平衡状態における応力の揺らぎと $G(t)$ は、Green-Kubo の関係式によって関係づけられ、応力の時間相関関数を計算することで $G(t)$ を求めることができる。応力の時間相関関数の計算には、長時間相関を効率的かつ精度よく計算できる multiple-tau correlator 法が有効であり、LAMMPS にこの手法が実装されている。一方、MD シミュレーション自体は GPU に最適化された HOOMD-blue を用いることで高速に実行できる。

しかし現状の実装では、HOOMD-blue (GPU) による MD 計算と、LAMMPS (CPU・MPI 並列) による multiple-tau correlator 計算を交互に実行する逐次処理を用いており、GPU 演算の終了後に CPU へ応力値を受け渡すまでの間、GPU がアイドル状態となる。このため、演算資源の利用効率に課題があった。

この問題を解消するため、Python の multiprocessing モジュールを用いた非同期並列実装（図 5）を新たに開発した。具体的には HOOMD プロセス（親プロセス）と LAMMPS プロセス（子プロセス）を独立したプロセスとして起動し、multiprocessing.Queue を介して応力テンソルの 6 成分をタプルとして受け渡す設計とした。

北海道大学情報基盤センターグランシャリオ 2 にて、Kremer-Grest モデル（鎖長 $N=100$ 、鎖本数 $M=100$ 、角度ポテンシャルあり $k=1.5$ ）を用いた性能評価を実施した。その結果、スループットが逐次処理版の 7,343TPS (timestep/second) から非同期処理版の 11,496TPS へと向上し、約 1.57 倍の高速化を達成した。また、求めた緩和弾性率 $G(t)$ について、逐次処理版と非同期処理版の結果が統計誤差の範囲で一致しており、非同期化による計算精度への影響がないことを確認した。

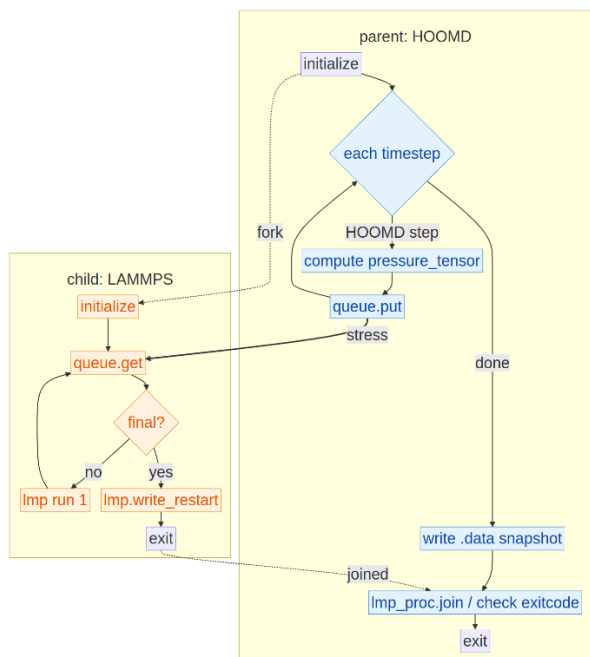


図 5 HOOMD-LAMMPS 非同期連携フローチャート

◆Tight Threading に関する粒子割合の結果

Tight Threading に関与しているかの確率 p_{TT} を、 $N = 100, 150, 200, 300$ の active polymer の系について評価した。予備的な結果は、 $p_{TT} = 0.04, 0.43, 0.47, 0.59$ であった。図 3 における $G(t)$ の値が長時刻領域でプラトーになる振る舞いと対応することを確かめた。今後より詳細を検討していく予定である。

[研究②] LAMMPS での分散 IO 技術の利用最適化～効率的な解析 platform とノウハウ形成～

Miyabi や富岳で実行される大規模な系の LAMMPS データ解析処理を効率よく実行する手法について検討した。大規模 MD 計算時の性能に加えて、データの移動を含めた解析における総合的性能も高くさせる「使い方ノウハウ」を形成することを目的とした。特に、HDF5 のシステムソフトウェアとしての VDS 機能を用いて、簡便かつ効率的にデータアクセスすることが可能かについて検討した。

LAMMPS を用いた大規模系の計算では、restart ファイル (バイナリ) を分散出力する。鎖同士のかみ合い解析を行う場合、鎖毎のデータを効率よく取り出したい。通常の利用では、分散出

力した restart ファイルを単一の Data ファイル (テキスト) に変換する。この処理でのメモリ確保が、大きなネックとなる。なお、Data ファイルや restart ファイルには、粒子の座標と、粒子同士の結合の情報が記録されている。

分散出力した restart ファイルへ直接アクセスする方法も考えられるが、本検討では、分散出力した restart ファイルの内容を、分散されたまま HDF5 に 1 対 1 変換し、HDF5 のシステムソフトウェアとしての VDS 機能で、見かけ上 1 つのファイルとしてアクセスする方法を検討した。なお、座標データは H5MD 形式を用い、トポロジーデータは独自 HDF 形式とした。

コード実装では、生成 AI を活用する時代の流れに合わせ、その活用も検討した。最初に、Restart ファイルから Data ファイルを出力させる python スクリプトを作成させた。具体的には、古いバージョンの LAMMPS (2013 年まで) に存在した restart2data.cpp という tool コードをリファクタリングし、2020 年版の LAMMPS のデータ形式に対応させた。生成 AI には、参考となるプログラムコード (旧版用の restart2data.cpp と、今回対象とする 5May2020 版の read_restart.cpp, write_restart.cpp, write_data.cpp 等) と試験用実データと正解の提供を含むプロンプトを与えた。当初は、ChatGPT 5 pro でコーディングできることを確かめ、その後の Codex CLI でも問題なくコーディングできることを確かめた。

座標とトポロジーのデータに、見かけ上単一ファイルでアクセスできるメリットはあるものの、実務上は、データ解析処理手法が定まっている場合は適切にデータを集約削減して効率を上げることが有効である。(その意味では、HDF5 に変換せずに、分散 restart ファイルから、直接、データを集約削減する方が良い場合も多い。) 少なくとも、ロスレスでのデータ圧縮という点では、HDF5 利用はメリットがある。KG 模型で 1 億粒子 (106,954,752atoms) の場合と 28 億粒子 (2,887,778,304atoms) の場合について、データサイズを評価した結果が表 1 である。

表 1 KG 模型に関する各データのサイズ

粒子数	1 億	28 億
Rst 合計	12.9GB	346GB
Text	21.1GB	未作成
HDF5(atoms)	5.3GB	141GB
HDF5(bonds)	0.5GB	13.0GB

今回は、HDF5 の VDS 機能の活用を焦点を当てて検討した。Data ファイル相当のデータに、HDF5 の VDS 機能を用いれば、分散した restart ファイルの 1 対 1 変換で、簡便かつ効率的にデータアクセスできることを確かめた。一方で、データ解析処理の実務面では、データ解析処理手法が定まっている場合、適切な処理ステップでデータを集約削減する方が効率的である。また、テストデータを適切に用意すれば、コード実装は、Codex CLI 等の生成 AI を活用してうまくできることを確かめることができた。すでに一般的普及が急速に進んでいるが、コーディングでの生成 AI 活用は、今後、ますます重要であることを認識できた。

6. 進捗状況の自己評価と今後の展望

[研究①]については、最も核となる Band 手術を活用した解析コードはまだ試作段階にあり、作業が若干遅れている。一方で、解析の対象とする MD シミュレーションのデータ蓄積は、おおむね順調に進んだ。また、解析を実施するためのインフラとして、MDX-1 や北大 K8S の利用方法の検討などは、概ね順調に実施することができた。

今後は、解析コードを完成させるとともに、絡み合い状態を詳細に区別するように解析手法を高度化する。なお、来年度以降は、本解析コードの開発・改良を担当する東北大 阪田先生が代表の JHPCN 課題(jh260052)で引き続き検討する。

[研究②]については、分散記録された MD データ (restart ファイル) へのアクセス性向上のため

に、HDF5 の VDS 機能を利用した python スクリプトを作成した。当初想定したことは、一定程度達成したが、実務上の効率としては具体的な解析に応じてデータを集約削減の方が明らかに良い。よって、この検討は、HDF5 での圧縮保存というメリットまででとどめておくこととした。

※7. 研究業績はウェブ入力です