

動的資源管理機能を備えた並列プログラミング環境の開発と 分子動力学シミュレーションへの応用

6 D-2

関嶋 政和 村田 達也 正木 宏和 中村 周吾 池口 満徳 清水謙多郎

東京大学大学院 農学生命科学研究科 応用生命工学専攻*

1 はじめに

現在、並列計算機あるいは分散システム上での並列処理を支援するためのプログラミング環境として、PVM, MPIなどのシステムがあり、一般に広く利用されている。とくに、分子シミュレーション、量子計算、流体計算など、大規模並列計算を必要とする様々な応用では、これらのシステムを用いてプログラムが記述され実行されることが非常に多い。しかしながら、これらのシステムは、プロセス間通信(メッセージ受渡し)機能の充実に重点が置かれ、十分な資源管理機能をもたないため、計算機資源を有効に利用し、並列プログラムの性能を向上させようという目的に十分応えることができないでいる。

とくに科学技術計算では静的負荷分散が困難な場合が少なくなく、そうした応用では動的負荷分散の機能が望まれる。し、分散システムをターゲットにした従来のプロセス単位の負荷分散はほとんど役に立たない。そこで本研究では、アプリケーションの構造に応じた動的資源管理機能を備えたプログラミング環境 Parsley の開発を行い、さらにその上で生体高分子シミュレーションを行った。Parsley は現在 HITACHI SR2201 と Sun Ultra Enterprise 10000 で稼働している。

2 並列プログラミング環境の構築

2.1 Parsley の基本構成

Parsley では図1のような階層構造によって構成される。本システムの特徴である動的資源管理機能は MPI の上に実現し、MPI では行なえないきめの細かな資源割当てを制御する。RMS (Resource Management Service) は基本的な資源管理を行うモジュールである。TMS (Task Management Service) はプロセスの割当てを制御するモジュールである。SMS (Subtask Management Service) は SPMD 形態に基

づく並列処理において利用されるもので、サブタスク割当て方針やタスクプール機構などをつかさどる。

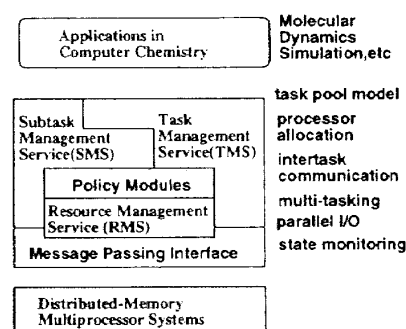


図 1: 並列計算のためのシステム構造

2.2 サブタスク間の依存関係の記述

アプリケーションプログラムは並列処理可能な処理の単位であるサブタスクの集合として構成される。サブタスク間に実行順序の制約(一方の結果を受けて他方が実行されるなど)が存在する場合には、ユーザはその制約を依存関係として記述する。

2.3 タスクプール方式の実現

Parsley における並列処理は図2のように、SPMD 形態のマスタ/スレーブ方式に基づいており、マスタがスレーブに依頼する形でサブタスクが実行されていく。実行されるべきサブタスク群はマスタでタスクプールを構成しており、依存関係を満たしたものから実行可能となり、さらにその中からスレーブの空きプロセッサに順次割り当てられる。現在、サブタスクの実行時間や通信の高速化などを考慮し、通信し合うサブタスク間の依存関係もユーザが指定できるようなきめの細かなプロセッサ割当て方針を検討している。

実行結果の受渡しについては、すべてマスタに返すのではなく、スレーブ上でサブタスクが直接受け渡すことにより、プロセッサ間通信の低減を図っている。この場合、スレーブはサブタスクの実行が終了すると、終了したことを伝えるメッセージだけをマスタに返し、スレーブは実行結果を一時的に保存するとともに、次のサブタスクの実行を開始する。他のプロセッ

*Parallel Programming Environment with Dependence-Driven Subtask Scheduling: Design and Application to Molecular Dynamics Simulation, Masakazu Sekijima Tatsuya Murata Hirokazu Masaki Shugo Nakamura Mitsunori Ikeguchi Kentaro Shimizu, Department of Biotechnology, The University of Tokyo

サで後続サブタスクの実行が開始されたとき、そのサブタスクに直接実行結果を送信する。

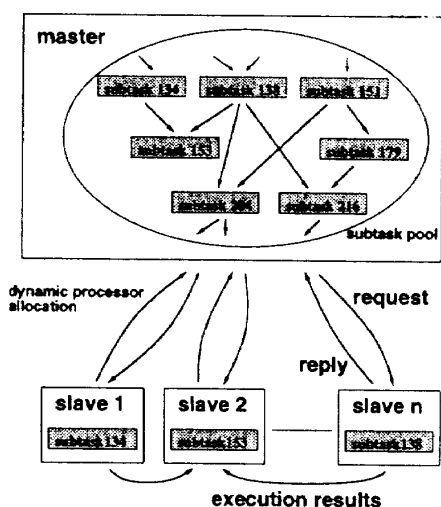


図 2: サブタスクの実行形態

3 生体高分子シミュレーションの実現

3.1 分子力学計算の概要

生体高分子の分子力学 (Molecular Dynamics, MD) シミュレーションを上記 Parsley の上に実現した。MD シミュレーションでは分子の動きを表すニュートン方程式を有限差分法を用いて解を求める。この解析によって分子の静的、動的特性を求めることが出来る。MD シミュレーションでは有限差分法のタイムステップごとに原子間に働く力の計算及び原子座標の更新を行う。

3.2 本機構における MD の並列化

並列計算に際しては、原子を空間分割の手法 [1] によりグループ化して、グループ内の原子に関する計算をサブタスクとして定義する。

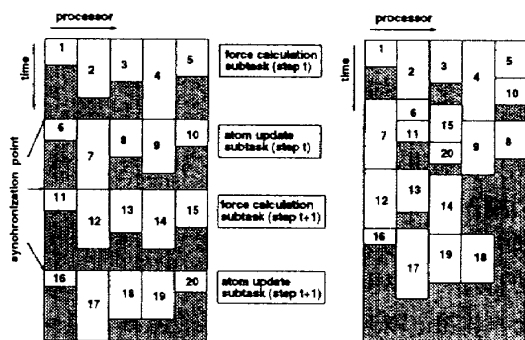


図 3: 通常の MD 並列化と Parsley による並列化

図 3は MD の並列化を通常の方法で行った場合と、

本システムで行った場合での比較である。通常の並列化の場合、分子にかかる全ての力の計算 (座標の更新) を行ってから座標の更新 (力の計算) を行うため、それぞれのステップの計算時間が最大なものに性能が依存してしまう。しかし、Parsley を用いることで、座標の更新 (力の計算) を行うのに必要な部分の力の計算 (座標の更新) をすればよくなるため性能が向上すると期待される。

3.3 結果

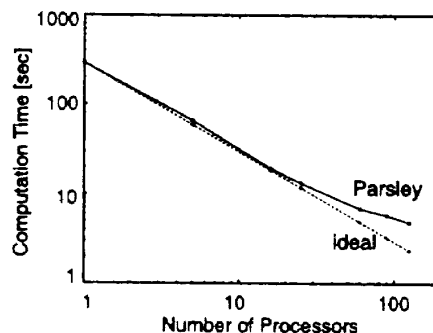


図 4: Parsley による MD の並列化効率

性能測定は、タンパク質 BPTI と水分子を合わせた系 (原子数 16735 個) で行い、空間を $5 \times 5 \times 5$ で分割した (図 4)。30 台程度のプロセッサ台数までスケラビリティが得られて得られていることが分かる。125 プロセッサに近付く程、スケラビリティが低下しているが、これはマスタの性能の限界ではなく、サブタスクの数がプロセッサの数に近付いたために並列度が低下しているからである。

4 結論及び課題

サブタスク間の依存関係に基づくタスクプール方式による並列計算法を開発し、生体高分子シミュレーションに適用して、その有効性を示した。

検討中の課題として、サブタスクグラフの自動更新、サブタスクの粒度の動的な調整、ポリシーモジュールの構造化設計、スケジューリング方針の検討、その他、高速化技術の検討として、マスタの多重化、サブタスクの動的切り替え、転送データの圧縮などがある。

現在、大規模分子の結合エネルギーの計算 [2] など、他の計算化学の応用に適用し、その有効性を調べている。

参考文献

- [1] P.Ballestrero, P.Baglietto and C.Ruggiero, *J.Comp.Chem.*, **17**, 469 (1996).
- [2] S.Nakamura, M.Ikeguchi and K.Shimizu, *J.Comp.Chem*, in press.