

論文の内容の要旨

論文題目 Parallel Programming with Tree Skeletons

(並列木スケルトンによる並列プログラミングの
理論と実現に関する研究)

氏名 松崎 公紀

計算機の能力の向上に従って、我々が扱うべきデータは非常に巨大になっており、また増加し続けている。そのような巨大なデータを効率良く処理するために並列計算はその重要性を増している。近年、高速なネットワークによって接続された PC クラスタやマルチコア・マルチ CPU のように並列計算を行うためのハードウェアは比較的簡単に手に入れることができるようになってきている。一方で、効率良く並列計算を行うための並列プログラムを作成することはそれほど簡単でない。並列プログラミングにおいては、データの分散、プロセッサ間の通信・同期、処理のスケジューリングなど、逐次プログラムにはない事項を考えなければならないからである。

並列プログラミングに関して、これまで多数のプログラミングモデルが提案されてきた。Cole によって提案されたスケルトン並列プログラミング (Skeletal Parallelism) はそのようなモデルのひとつである。スケルトン並列プログラミングでは、並列スケルトンと呼ばれる並列計算に頻出する抽象的な計算パターンを組み合わせることで並列プログラムを作成する。データの分配やプロセッサ間通信などの並列性が並列スケルトンの中に隠蔽されることにより、ユーザは比較的簡単に並列プログラムを作成することができる。これまでに、リスト (配列) に対して多くの研究があり、様々な優れた結果が得られている。

本論文は、木構造の処理を抽象化した並列スケルトン (並列木スケルトン) を用いた並列プログラミングについて、その理論と実現に関する研究をまとめたものである。

木は構造を持ったデータを表現するのに良く利用される重要なデータ構造である。しかし一方で、木は不規則・不均等な構造を持つため、木上のデータを効率良く並列に処理することは困難であった。例えば、木を処理する多くの逐次アルゴリズムは分割統治法によって記述されるが、そのアルゴリズムをそのまま分割統治法による並列アルゴリズムにただけでは入力の木が不均等な場合に性能が悪くなる。この問題を解決した、任意の形状の木に対して効率が保証できる **tree contraction** アルゴリズムと呼ばれる並列アルゴリズムがある。しかし、この並列アルゴリズムでは計算の順序が特殊であるため、もとの逐次プログラムを相当に変更しなければならないことが少なくない。また、この並列アルゴリズムを分散メモリ並列計算機上で効率良く実現することは簡単ではない。これらの理由から、計算を抽象化した並列スケルトンによるスケルトン並列プログラミングは木に対する並列処理を簡易にするための魅力的なプログラミングモデルである。

本論文の貢献は、並列木スケルトンによるプログラミングの理論と実現の両面からなる。理論に関する研究では、構成的アルゴリズム論 (**Constructive Algorithmics**) に基づく並列木スケルトンの設計、および、それらの並列木スケルトンを用いた系統的なプログラミング手法を提案した。実現に関する研究では、分散メモリ並列計算機においても効率の良い並列木スケルトンの実装を含む並列スケルトンライブラリ「助っ人」の実装、および、ふたつのクラスの自明でない実用的な問題に対するスケルトン並列プログラミングの適用を行った。以下に本論文における重要な4つの貢献をまとめる。

第一の貢献は、構成的アルゴリズム論に基づく並列木スケルトンの設計である。構成的アルゴリズム論はデータの構成とその上のアルゴリズムの関係に着目した理論であり、逐次プログラムの設計・導出に関して多くの成果がある。並列木スケルトンは **Skillicorn** によって最初に定式化されたが、本論文において二分木と一般の形状の木のそれぞれに対して並列木スケルトンを再設計した。これらの並列木スケルトンは、再帰関数による逐次的なインターフェイスと様々な並列計算機環境における効率良い実現というふたつの側面を持つ。一般には、任意の計算が並列に計算できるわけではないため、いくつかの並列木スケルトンは補助関数を要求する。この補助関数を用いた定式化は本研究の新規性のひとつであり、これらの補助関数はプログラム導出や効率の良い並列実装において重要な役割を果たす。

第二の貢献は、系統的なスケルトン並列プログラミング手法の提案である。並列木スケルトンを利用することで多くの並列プログラムを作成することができるが、ユーザの視点に立つと、それらをどのように組み合わせるか、また、要求される補助関数をどうやって見つけるかを誘導することが必要である。まず、並列木スケルトンの組み合わせ方を導出するために分離定理 (**Diffusion 定理**) を示した。この定理によって、逐次の再帰アルゴリ

ズムがどのような並列スケルトンの組み合わせに分解できるかが示される。また、並列木スケルトンに要求される補助関数を導出するために計算の代数的な性質に着目した。本論文で示す 3 種類の代数的性質のいずれかが成り立つ場合には、系統的もしくは自動的に必要な補助関数を導出することができる。

第三の貢献は、並列スケルトンライブラリ「助っ人」の実現である。並列スケルトンライブラリ「助っ人」は汎用的な C++ 言語と MPI ライブラリを用いて実装されており、分散メモリ環境の上でも効率の良く動作する並列スケルトンを提供する。特に、リストや行列を扱う並列スケルトンだけでなく、木を扱う並列木スケルトンが実装されていることは、他の並列スケルトンライブラリにはない「助っ人」の重要な特徴である。「助っ人」のもうひとつの重要な特徴は、融合変換による最適化機構を持つことである。融合変換による最適化機構を利用することで、多数の並列スケルトンによるオーバーヘッドを取り除くことが可能である。並列木スケルトンの実装や最適化機構の効果について、実際の PC クラスタの上で実験を行い検証した。

第四の貢献は、自明でない実用的な問題に対するスケルトン並列プログラミングの適用である。本論文では、最大マーク付け問題（最大重み和問題）と XPath クエリ処理とのふたつのクラスの問題が、スケルトン並列プログラミングによってそれぞれ統一的に並列化することができることを示した。最大マーク付け問題には動的計画法で解ける最適化問題が含まれており、リストや行列に対しては多数の研究がある。木における最大マーク付け問題の一般的な並列化を示したのは、本論文が初めてである。これらの並列プログラムの導出においては、第二の貢献である系統的なスケルトン並列プログラミング手法が重要な役割を果たしている。

以上が本論文の要旨である。