

外92-53

早稲田大学大学院理工学研究科

## 博 士 論 文 概 要

### 論 文 題 目

技術計算プログラムに対する  
並列処理方式の研究

申 請 者

山本富士男

FUJIO YAMAMOTO

平成 4年 12月

理 1650 ( 1939 )

本研究は、科学技術計算分野において、ベクトル計算機および並列計算機の高性能を効果的に利用するためのソフトウェア技術の開発を目的としたものであり、「大規模疎行列高速解法の研究」と「並列記述言語とその処理方式の研究」の2つからなる。

科学技術計算において、コンピューティングシステムの能力に対する要求は、その解析の精密化と適用分野の拡大に伴って益々高度なものになりつつある。これに対するソフトウェア技術面の課題は大きく分けて2つあると考えられる。第一は、数値シミュレーションの高速化技術であり、使用計算機の高性能を十分に利用するには、ベクトル演算や並列演算などの利用による高速化を強く意識して計算アルゴリズムを設計することが不可欠となっている。第二の課題は、このような状況での応用プログラム開発の生産性や保守性の向上である。例えば並列計算機向けプログラミングは、プログラムの分割配置問題や分割された処理間の同期問題など、単一計算機の場合には無い困難さが伴うので、その生産性向上は基本的に重要である。また、今後多様な並列アーキテクチャが次第に普及する情勢にあるので、プログラムの移植性や保守性の高さが求められる。

本論文は上記の2つの課題に対する研究を述べたものであり、全体で7つの章から構成される。第1章は本研究の背景と目的を述べている。ここでは、ベクトル計算機および並列計算機の発達に伴って生じたソフトウェア技術面の課題を考察し、それに対して行なった研究の過程も述べた。

第2章と第3章は、最初の課題に対応する「大規模疎行列高速解法の研究」について述べたものである。第2章では、数値シミュレーションの典型的の一つである、大規模回路シミュレーションを取り上げ、そのベクトル型スーパーコンピュータ上の高速化技術を検討した。従来計算時間の大半を占めていた、直接解法による線形方程式求解は、その係数が不規則疎行列であるため、当時出現し始めたベクトル型スーパーコンピュータ上の高速化は困難と言われていた。これに対し筆者は、2つの新たなアルゴリズムMVA (Maximal Vectorization Algorithm)とBVA(Block Vectorization Algorithm)を開発した。両アルゴリズムとも、不規則に散在する非零要素に対する演算の並列性を抽出するものである。MVAは、非零要素に対する定義と使用の状況を表現した「データ参照レベル」に基づいて行列全体に渡って並列性を最大限に検出する。一方、BVAは並列性の検出範囲を連続した列ブロック内に限定する代わりに、一般に稠密となる右下隅小行列部に対する高速密行列解法への動的切り替えを可能にする。これらの手法は、一回のベクトル演算での処理データ量、すなわちベクトル長を従来法より数十倍長くし、ベクトル化による加速率を格段に高めた。実際、2200トランジスタ規模の実回路の過渡応答解析時間を従来法より約1桁短縮することができた。

第3章では、さらに高い高速化の可能性を求めて反復系の行列解法を検討した。回路シミュレーションにおいては、全体回路を部分回路に分割し、それら部分回

路レベルの信号伝播をガウスザイデル法などの反復法で解くWaveform Relaxation法等があったが、現実のメモリ回路等に適用するためには、最適分割法および安定した収束性確保など研究課題が多く残されていると考えられた。そこでここでは、これまで高い信頼度で多用されてきた、全体回路行列をそのまま解くという立場での行列反復解法を検討した。しかし、偏微分方程式系の解法には有効であった行列反復解法は、回路行列のような不規則疎行列にはほとんど使用されていなかった。これは、この不規則性のため、行列固有値を密集させて収束性を高めるための効率的な前処理法(preconditioning)が無かったことに起因する。これに対し、筆者は、回路の過渡応答解析の特性から、ある時点の回路行列に対する完全なLU分解の結果は、それ以降のかなり長い時間区间に渡って行列前処理の係数行列として有効に利用できることを実験的に突き止めた。さらに、反復の収束判定を従来の残差だけの観測によるのではなく、解の修正活動状況の多面的な観測に基づいて行なうことで、次に完全LU分解が必要となる時期を適切に検出できる方式を考案した。これらにより、現実のDRAM回路などでも安定に収束解が得られることを確認した。また、その分析結果から、約6000元以上の大きな回路行列では、従来の直接解法の性能を凌駕するという予測を得た。

第4章～第6章では、第2の課題に対応する、「並列記述言語とその処理方式の研究」を述べている。ここでは、上記の高速行列解法を一例とするような応用プログラムを、今後の並列計算機向けに効率的に開発するための手段を提供することを狙った。まず第4章では、「並列言語仕様の策定」について述べている。技術計算分野の並列記述言語としては、FORTRAN90がその最終仕様決定の段階にあった。筆者らは、それに採用された配列型並列性記述機能だけでは、一般的な技術計算の並列処理記述に不十分であると考え、多くの応用問題から要請される記述機能を分析して新たな言語仕様を定めた。この言語の特徴機能として、計算手続きの並列処理を簡潔に表現するためのブロック並列記述機能と疎行列記述機能を策定した。特に疎行列記述機能は、現実の応用問題に多く出現する各種の疎行列構造の定義とそれに対する演算の簡潔な表現を可能とし、従来の密行列に対する配列型並列記述の自然な拡張を図ったものである。また、粒子系の問題に対する、粒子の初期速度や位置の分布などの属性記述機能等も付加し、応用問題解法の記述性の向上を図った。離散系および粒子系の実Fortranプログラム4題（合計15Kステップ）を並列化し本言語で記述した結果、元のFortranの記述量の38～77%（平均56%）で済むことが分かった。

第5章では、第4章の並列言語で記述された応用プログラムを、並列計算機にマッピングする言語処理ソフトウェアと、その処理結果として得られた応用プログラムの実行時性能について述べている。マシンとしては筆者が参加していた、通商産業省工業技術院大型プロジェクト「科学技術用高速計算システムの研究開発」の並列計算機を対象とした。言語処理ソフトウェアの内で特に、逐次ループ、

並列ループ、および並列ケース（ループでは表現できない、異なる種類の処理単位が並列に実行可能な構造）が混在する並列構造へプロセッサを割り付ける方法を検討した。当時このような構造、すなわちプログラムの並列構造を表すタスクグラフのノード自体が並列ループである場合のプロセッサ割り付け法は他に報告されていなかった。これに対し筆者は、最急降下法の考え方に基づく並列ケースの各分岐へのプロセッサ台数の配分の最適化方式を考案し、これを従来の多重ループ構造への割り付け法と融合させた。これは、プロセッサ台数に対する実行時間評価関数の勾配ベクトルを、プロセッサ台数という自然数の空間で近似的に利用するものであり、並列ケースのどの分岐へプロセッサ割り付け台数を1台増やすと実行時間評価が最短になるかを計算してプロセッサ台数配分を順次決定していく点に特徴がある。数例の実プログラムに対し、4台プロセッサ構成の上記並列計算機を用いて、本言語処理ソフトウェアが自動生成したコードの実行時性能を評価し、その効果を確認できた。帶行列の並列解法問題では1台プロセッサの場合に比べ、4台で2.83倍、また固体物理の問題では4台で3.78倍の高速化を実測した。

第6章では、この種の言語による並列プログラムを、分散記憶型並列計算機にマッピングする場合の課題の考察と、データの分割の仕方の評価方法を述べている。近年、比較的安価な小型プロセッサ多数台をネットワークで結合した形態の並列計算機が台頭してきた。その多くは、各プロセッサが固有の記憶領域をもつ分散記憶型であるため、言語処理ソフトウェアの最大の課題は、プログラムのデータと処理を如何に分割して高性能プログラムを生成するかにある。筆者は、それに対する重要な研究として、データの分割方法を組織的に評価する方式を検討した。これは、予め定めた列分割、行分割など基本的なデータ分割関数（2次元配列の場合は6種）をプログラムに出現する各配列に適用して得られる、データ分割方法の各候補を評価するものである。すなわち、その各候補に基づいてプログラムの機能を分割した場合の各プロセッサの計算量の均衡度、通信回数、通信データ量などを相対的に評価するプログラムを自動生成する。これを実機使用前に実行することにより、事前に良いデータ分割を得ることを狙った。一例としてガウス消去プログラムに対し、16台プロセッサを想定した6種のデータ分割候補の中から本方法によって最適なものを見出した。それは、筑波大学の並列計算機PAX上の性能実測値に基づく最適データ分割と一致した。

第7章は結論であり、本研究で得られた結果を総括している。

以上に述べた研究により、現在実用化されているベクトル型スーパーコンピュータと、今後益々その需要が拡大すると予測される並列型スーパーコンピュータを効果的に利用するため技術開発に貢献できたと考える。前半の研究の手法は、米国クレイ社など5社で採用され、大規模回路解析等における有効性が実証されるに至った。また、後半の研究では、並列プログラミングの困難さを緩和するための一つの方向を提案し、その実用面での可能性を示すことができたと考える。