早稲田大学大学院理工学研究科

博 士 論 文 概 要

論 文 題 目

High Performance Fortran 処理系の
設計、実装および評価

Design, Implementation, and Evaluation of
a High Performance Fortran Compiler

申 請 者
太 田 寛
HIROSHI OHTA

2000年12月

スーパーコンピュータ（以下「スーパーコン」）は、気象予測、遺伝子解析、経済解析、建造物設計、材料設計、自動車や航空機の設計等の様々な分野で利用されている。これらの分野で必要とされる大規模問題を高速に解くためには、通常の計算機の数十倍から数千倍の計算能力を有するスーパーコンが不可欠であり、現代社会を支えるために必須の道具となっている。その性能は年々向上しており、２０００年現在で、実効性能が１テラフロップス（１秒間に１兆回の演算を実行する能力）を超えるものもいくつか稼動している。

このような高性能を達成するために、スーパーコンのアーキテクチャも変遷している。１９８０年代までのスーパーコンは単一の強力なベクトルプロセッサを備えるベクトル型が主流であったが、１９９０年代になって、多数のプロセッサを結合した並列型が主流となった。さらに、並列型のスーパーコン（以下「並列スーパーコン」）の中でも、共有メモリ型から分散メモリ型への移行が起こっている。すなわち、初期の並列スーパーコンではプロセッサ数が少なかったため、１箇所に集中して配置された主記憶を複数のプロセッサが共有する集中共有メモリ型構成が用いられてきたが、プロセッサ数が数十台以上に増加するにつれて、複数プロセッサからのメモリアクセスの競合を回避するため、メモリを各プロセッサに分散配置する分散メモリ構成が用いられるようになってきた。

このようなアーキテクチャの発展によって並列スーパーコンの性能は向上したが、一方でユーザに対してプログラミングの負担を強いることになった。現在は、分散メモリ構成の並列スーパーコン（以下「分散メモリマシン」）のユーザの多くは、「メッセージ通信型」のプログラムを書いている。これは、各プロセッサ毎の計算処理や他プロセッサとの通信を、明示的に記述したプログラムである。メッセージ通信のための代表的なライブラリ仕様としては、Message Passing Interface（MPI）が知られている。メッセージ通信型プログラム内では、そのプロセッサが担当する計算の範囲を求めたり、通信すべきデータ集合を求めて適切な順序で送受信するなどの処理が必要となる。このような処理の記述はユーザにとって面倒な作業であり誤りもおかしやすい。

このユーザの負担を軽減するためのプログラミング言語として、High Performance Fortran（HPF）が提案されている。HPFは、Fortran言語にデータ分散配置などの並列化指示文を追加した言語であり、米国を中心とした大学や企業の代表から構成される団体 HPF Forumによって仕様が策定された。HPFを用いた場合、ユーザは各プロセッサ向けのプログラムを書くのではなく、逐次処理の場合と同様に全体としての計算処理を記述する。そして、指示文によって基本的には各プロセッサへのデータ分散のみを指定する。HPF処理系は、このデータ分散指示文に基づいて、自動的に計算処理の割当てや通信生成を行う。ユーザは、各プロセッサの計算範囲や通信集合を気にすることなく、プロセッサ全体としての計算処理のみを記述すれば良いので、プログラム作成が容易になるという利点がある。

しかし一方で、従来は HPF 処理系の技術が未熟であったため、複雑なパターンの HPF プログラムが入力された場合に、高性能なコードが生成できず十分な実行性能が得られないという問題があった。

本論文の目的は、この問題を解決するための処理系技術を提案し、現実のプログラムに現れる様々な入力パターンに対して高性能なコードを生成できる HPF 処理系を開発することである。

本論文は６章から構成されている。

第１章では、序論として本論文の背景と目的を述べる。

第２章では、処理系技術の一つである計算分散技術、特に従来十分に焦点が当てられて来なかった多重ループの計算分散について述べる。多重ループには、配列のある次元の添字がループ制御変数を含まない場合や、配列の各次元の添字がループ制御変数と１対１に対応しない場合など、単なる１重ループの組み合わせでは対処できない様々な場合が存在する。このような場合、従来の処理系はコンパイル時の計算分散をあきらめて、「実行時解決法」に基づくコードを生成してしまうという問題があった。これは、全プロセッサが元のループの繰返し範囲全体を実行し、配列参照の度に毎回、参照される要素が自プロセッサに存在するかどうかを判定するというものであり、非常に実行性能が悪い。

本論文ではこの問題に対して、「マッピング標準形」を用いた統一的な計算分散方式を提案する。マッピング標準形はデータ分散や計算分散を表現するための表現形式であり、配列の形状や分散次元などを示すいくつかのパラメタによって構成される。そして、分散された配列のマッピング標準形と配列添字から、計算分散に対するマッピング標準形を決定するアルゴリズムを示す。さらに、計算のマッピング標準形に基づいて、元のループを各プロセッサ毎の処理を表すループに変換するアルゴリズムを示す。

以上の計算分散方式によれば、配列の分散次元添字がループ制御変数の線形式で表される場合には、少なくとも一つのループがコンパイル時に計算分散され、実行時解決法が回避できるようになる。本方式を様々な配列添字を含むループに適用し、分散メモリマシン日立 SR2201 で評価した結果、プロセッサ数が４台から 16 台の範囲において、実行時解決法と比べて 1.4 倍から 2.9 倍の性能向上が得られ、本方式の有効性が示された。

第３章では、通信生成技術について述べる。通信生成の一手法として、配列の再マッピングを利用する手法が従来から提案されていたが、本章では、この手法を前述のマッピング標準形を用いて一般化し、かつ最適化強化を行うことにより、様々な配列添字を持つループへの適用性を高める。マッピング標準形を用いることにより、シャドウ通信や１対１通信などの高速通信パターンの認識が容易になるという利点もある。

本章の通信生成手法を、並列スーパーコンの代表的ベンチマークの一つであるMG(マルチグリッド法)に適用して評価した結果、従来の他の研究においてHPF版はMPI版と比べて数倍の実行時間を要していたのに対して、本手法の適用により、SR2201の16プロセッサ時においてHPF版の実行時間をMPI版の1.18倍にまで短縮できることが示された。すなわち、メッセージ通信を用いた人手並列化と比べて遜色のない性能が、HPFによって得られることが示された。

第4章では、通信生成の中でも特に、ループ運搬依存がある場合のDOACROSS型通信の最適化技術について述べる。DOACROSS型の多重ループでは、通信起動オーバーヘッドを低減するため、複数のループイタレーションをまとめた「タイル」を単位として通信を行う「タイリング」技法が知られている。このとき、タイルサイズを大きくすれば通信回数は減るが、2番目以降のプロセッサの処理の開始が遅れるというトレードオフがある。

本章ではこのトレードオフを考慮して、一般的な依存ベクタを持つループに対するタイリング手法、特に最適タイルサイズの決定に関する理論を示す。分散メモリマシンnCUBE2上で、様々なDOACROSS型多重ループに対してタイルサイズを変化させて評価した結果、実測で得られた最適タイルサイズと理論予測値とは誤差25%の範囲内で一致し、理論の有効性が示された。

第5章では、前章までに述べた処理系技術を実装したHPF処理系を、並列スーパーコンの代表的なベンチマークセットである NAS Parallel Benchmarks (NPB)によって評価した。NPBをHPFで記述する試みは従来からも行われていたが、8本のベンチマーク全てを対象とはしておらず、また実行性能も不十分であった。

本章ではNPBの全8本についてHPF版を作成し、前述のHPF処理系によって並列化した。この過程において、前章までの処理系技術が、2本のベンチマーク（MG、LU）において必須であることを示した。また、高い実行性能を得るためのプログラミング技法として、再分散（データ分散の動的変更）の活用や、手続き呼び出しを含むループの指示文による並列化などが重要であることを明らかにした。このHPF版NPBを前述のHPF処理系で並列化しSR2201上で評価した。その結果、プロセッサ数8台を基準として、台数を2倍（16台）、4倍（32台）、8倍（64台）にしたときの性能向上が、8本の平均でそれぞれ1.83倍、3.15倍、5.02倍という値が得られ、良好なスケーラビリティが達成された。

第6章はまとめであり、前章までに得られた本研究の成果を総括する。