早稲田大学大学院理工学研究科

博 士 論 文 概 要

論 文 題 目

細粒度並列アーキテクチャのための

ループ並列化技法

申 請 者

古関 聰

Akira Koseki

電気工学専攻 計算機ソフトウェア研究

1997 年 12 月

半導体技術の発達により 1 チップに数十万個のゲートが集積されるようになり、1つのプロセッサの中で同時に機械語を実行するスーパースカラ、VLIW などの命令レベル並列プロセッサが開発、実用化されてきた。特にスーパースカラプロセッサは、Pentium、PowerPC、SuperSPARC、Alpha などのチップが開発され、ワークステーション、パーソナルコンピュータに広く搭載されている。これらのプロセッサでは、一度に 2~8 個の機械語が同時に実行されており、現在も同時実行可能な命令数を増やす研究が行われている。また、これらの命令レベル並列プロセッサの研究・開発と同時に、複数個のプロセッサを相互に接続して並列処理を行うマルチプロセッサの研究が行われてきた。相互結合の方式によってマルチプロセッサはいくつかの方式に分類されるが、中でも、各プロセッサが共通にアクセスできる共有メモリを持ち、このメモリの読み書きを介して通信を行う共有メモリ型並列計算機に大きな期待を寄せられている。すでに、Pentium、SuperSPARC などのプロセッサを 2~8 個搭載し、共有メモリを介した通信を行いながらの並列処理が可能な計算機が実用化されており、さまざまな分野でのハイエンドコンピューティングに活用されてきている。

これらの並列計算機を有効に活用するための重要なコンパイラ技術にループ最適化がある。ループ最適化とは、ループの繰返しの間にある並列性を抽出し、各処理体（命令レベル並列プロセッサでは各演算装置、マルチプロセッサでは各プロセッサ）が同時に実行可能なコードを生成することである。ループ最適化は、並列実行可能な命令を抽出する方法により、以下の二つに分類できると考えられる。

1. データ並列性（空間的並列性）を利用したもの
2. パイプライン並列性（時間的並列性）を利用したもの

前者は、異なる繰返し中の同一命令（たとえば、ループの繰返しの 1 回目における命令 A と 2 回目における命令 A）が異なる処理体で同時に実行されるように、プログラムを変形する方法である。また、後者は、同一の処理体が異なる繰返し中の同一命令を実行するように、プログラムを変形する方法である。例えば、命令レベル並列プロセッサのためのデータ並列型ループ並列化技法としてループアンローリング、パイプライン並列型ループ並列化技法としてソフトウェアパイプラインが挙げられる。また、マルチプロセッサのためのデータ並列型ループ並列化技法としてイタレーション分割、パイプライン並列型ループ並列化技法としてループディストリビューションが挙げられる。

これらのループ並列化技法は、数値計算などのアプリケーションに対し非常に有効に適用できる。このようなアプリケーションのループにおいては、ループの各繰返し間に依存関係がないので、データ並列性、パイプライン並列性を自由に抽出することができるからである。これらのループは DOALL ループと呼ばれる。しかしながら、多くの実用的なループではループ並列化技法を単純に適用することはできない。また、プログラムの性質によっては、まったくループ並列化の効果が期待できない場合もある。これは、繰返し間の依存関係によって命令間の並列性がなくなってしまっているからである。このような性質を持つループは DOACROSS ループと呼ばれており、DOACROSS ループ

をどのように並列化するかが、現時点でのループ並列化の大きな問題となっている。

そこで、本研究では、命令レベル並列プロセッサと、共有メモリ型マルチプロセッサを対象とした DOACROSS ループの並列化を試みた。具体的には、

1. 命令レベル並列プロセッサのための、データ並列性を利用したループ並列化技法の開発
2. 共有メモリ型マルチプロセッサのための、パイプライン並列性を利用したループ並列化技法の開発

を行った。

まず、命令レベル並列プロセッサを対象とし、データ並列性を利用したループの並列化を行うため、ループアンローリング技法の新しい適用アルゴリズムを開発した。

ループアンローリングはループを展開することによって、ループの繰返し間に存在する並列性を引き出す方法である。この方法では、展開後のプログラムに対して，共通部分式の削除や，配列変数を介したデータの授受を、スカラ変数を介したものに置き換えてロード／ストア命令を減らすスカラリプレースメントなどの最適化技法を適用することが可能である．さらに，定数回繰り返されるループでは，ループを展開することで，ループ脱出のための比較命令やジャンプ命令を削減することができる．これにより，命令をより連続的に命令パイプラインに投入できる．また，分岐を削減することでコードスケジューラによる最適化の制約が小さくなり，より並列性の高いコードを生成することができる．

しかしながら、ループアンローリングには、プログラムの性質とハードウェアリソースに合わせた最適な展開数でループアンローリングを行なわないと，展開数が足りずに十分な並列度が得られなかったり，展開数が多過ぎて効果は飽和しているのに命令数だけが多くなってしまうという問題点がある。

厳密な最適解を得るためには，ループを様々な回数で展開し，実際のコード生成を行なった後に、それらの実行速度を比較し最も効率のよいものを選ぶ必要がある．しかし，これは，解を得るまでに膨大な時間が必要であり，現実的な方法ではない．そこで，我々はループアンローリングの効果の飽和や低下を引き起こす様々な要因を考慮したヒューリスティクスを用い，適当な展開数を決定する方法を開発した。

具体的には、まず、ループアンローリングが行われる過程を分析し、展開数を n としたときのハードウェア制約による実行時間下限と、ソフトウェアの性質によって決められる実行時間下限を、ともに n の一次関数で近似した。ループアンローリングの効果が飽和する状態は、前者が後者を上回った状態に他ならないので、これらの一次式を連立させた不等式を解くことで、求める展開数（飽和がもたらされる展開回数）を得ることができる。

このヒューリスティクスを用い実験を行ったところ、評価プログラム中の 83%で最適展開数が得られた。また、残りについても、十分に妥当な展開数が得られていることが実証された。

次に、このヒューリスティクスの拡張として、ループアンローリングを多重ループにおける複数のループに同時に適用する試みを行った。以前の最内ループのみを対象としたループアンローリングでは、プロセッサの並列度を十分に満たすほど並列性が抽出できない場合がある。この研究によって、最内ループのみを展開した場合に比べ多くの並列度を引き出すことが可能になる。また、さらに重要な点として、近隣の繰返し間でのデータを再利用し、より多くの無駄なロード命令を減らせるようになる。

この研究の具体的な目的は、最内ループの場合と同じく、ループの適切な展開数を求めることである。ただし、多重ループの場合は、多重ループを構成する各ループに対し適切な展開数を求める必要がある。

展開数を求めるアルゴリズムは、最内ループの場合の自然な拡張になっている。主な違いは、ハードウェア制約による実行時間を求める式が、命令の再利用を考慮しているのでn次式になる点にある．このため、最適な展開数は、代数的に解くことはできない。本論文では，山登り法による最適展開数算定技法を開発し、評価プログラムの全てにおいて最適展開数が得られることを確認した。

最後に、ループステージングと呼んでいる、共有メモリ型マルチプロセッサを対象とした、ループのパイプライン並列性を利用したループ並列化アルゴリズムを開発した。

マルチプロセッサを対象としたループ並列化の研究は様々なものがあるが、その多くはループの繰返し間の並列性を利用したものである。しかしながら、繰返し間の依存性が静的に解析できないループ、あるいは、依存関係が多くて複雑なループでは、これらのループ並列化をうまく適用することができない。

ループステージングでは、ループ本体を構成している命令群を、ステージと呼ばれるいくつかの実行段階に分割する。そして元のループから、それぞれのステージを本体としたループを作り、このループ単位でプロセッサへの配置を行う。すなわち各プロセッサは担当するステージのループのみを実行する。このような命令分割を行うことで、各プロセッサの持つループをオーバラップさせ、パイプライン並列を利用して実行させることが可能になる。

このようなループ並列化を行うことで、静的な依存解析が困難である、バンド行列を介して行列演算を行うループや、繰返し間にデータ並列性のないシングルループを対象として、2〜4倍の速度向上を得ることに成功した。

以上に述べたように、ループ最適化によって、並列計算機の並列処理能力を活用することが可能である。特に、本論文が導入したアルゴリズムにより、命令レベル並列計算機、共有メモリ型並列計算機上でのアプリケーションの実行に多大な速度向上をもたらすことが確認された。