

スライドウィンドウを考慮したレジスタ割り付け

2N-7

添野 元秀 山下 義行 中田 育男
筑波大学

1 はじめに

超並列計算機 CP-PACS[4] の各ノードプロセッサは、多数の浮動小数点レジスタとそのレジスタ群に対するスライドウィンドウ機構 [1] を付加した RISC プロセッサを採用している。

当研究室では、スライドウィンドウを付加したプロセッサをターゲットマシンとするコンパイラの研究を行っている。コンパイラにおける最適化の重要課題の一つにループの最適化がある。RISC プロセッサにおけるループの最適化には、ソフトウェア・パイプラインが有効である。スライドウィンドウを用いると、スライド命令によってレジスタ番号が変化するため、ソフトウェア・パイプラインにおける複数のステージにまたがるレジスタの干渉が回避でき、命令スケジューリングの自由度が増すという利点がある。しかし、レジスタ番号が変化することは、レジスタ割り付けに新たな制約条件を加えることになり、レジスタ割り付けを難しくすると考えられてきた。

本稿では、スライドウィンドウを考慮していないレジスタ割り付けのアルゴリズムを比較的容易にスライドウィンドウを考慮したレジスタ割り付けアルゴリズムに拡張できることを述べ、そのアルゴリズムの評価を行う。

2 スライドウィンドウ

RISC プロセッサは、高いキャッシュヒット率を仮定してピーク時性能を実現している。しかし、CP-PACS の主目的である計算物理学で用いられるデータは、汎用性・局所性が少ないため、キャッシュヒット率が低下し、実行時性能が著しく低下する。スライドウィンドウは、多数のレジスタを既存の RISC プロセッサとの互換性を維持しながら実現したものである。物理レジスタを複数のレジスタウィンドウに分割し、このウィンドウをスライド命令でスライドさせながら計算を行うことで、メモリアクセスのオーバーヘッドが軽減される。

スライドウィンドウ構成を、図 1 に示す。

3 ソフトウェア・パイプライン

本研究で用いる最適化手法は、ソフトウェア・パイプラインを基本としている。これは、ループの命令列を複数のステージに分割し、それぞれのステージをパイプライン的に同時実行するものである。その実現には、各命令の実行のタイミングと変数へのレジスタ割り付けの制約問題を解く必要がある。スライドウィンドウを用いると、論理レジスタ番号が動的に変化するため、本研究では、まず命令スケジューリングを行い、その後で、レジスタ割り付けを行うアルゴリズムを採用する。

3.1 命令スケジューリング

通常のソフトウェア・パイプラインでは、複数のパイプライン・ステージで参照される変数の干渉が命令スケジューリングの制約となる。スライドウィンドウを用いると、スライド

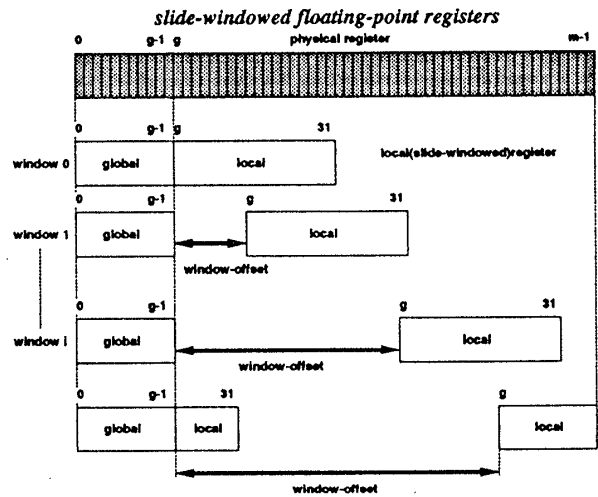


図 1: スライドウィンドウ構成

命令によってレジスタの番号が自動的に変化し、干渉が回避できるので、スケジューリングのアルゴリズムは自然な形で実現できる。

3.2 レジスタ割り付け

スライドウィンドウを用いると、レジスタ番号がスライド命令ごとに変化することになる。そのレジスタ番号の変化を新たな制約条件として、レジスタ割り付けアルゴリズムを拡張する。

従来のレジスタ割り付けでは、各変数のライブレンジをグラフのノードとし、ライブレンジが重なっているノードを枝で結んだ干渉グラフを用いて、グラフの彩色問題として割り付けるのが一般的である [2]。しかし、干渉グラフの彩色アルゴリズムは、必ずしも最適解を与えるわけではない。そこで、各種のレジスタ割り付けアルゴリズムが提案されているが、その中に、サイクリックインターバルグラフを用いるファットカバーアルゴリズム [3] がある。本研究では、干渉グラフを用いるアルゴリズムとサイクリックインターバルグラフを用いるアルゴリズムのそれぞれについて、スライドウィンドウを考慮したアルゴリズムに拡張する。

4 レジスタ割り付けアルゴリズムの拡張

4.1 干渉グラフを用いるアルゴリズム

4.1.1 グラフの拡張

干渉グラフでは、変数の定義点或使用点・消滅点の情報を表すことができない。スライドウィンドウを考慮する場合は、ウィンドウのスライド命令でレジスタ番号が変化するため、その情報をグラフに反映しなければならない。

本研究では、レジスタ番号が変化する変数を、変化前と変化後で別のレジスタにのると考え、それぞれのノード間にレジスタ番号の連続性を示す矢印をひくことで、スライド命令によるレジスタ番号の変化を表すものとする。

Register Allocation algorithms for Pseudo Vector Processor based on Slide Window Registers
Motohide Soeno, Yoshiyuki YAMASHITA, Ikuo NAKATA
(Univ. of Tsukuba)

スライドウィンドウを考慮した干渉グラフの例を図2に示す。

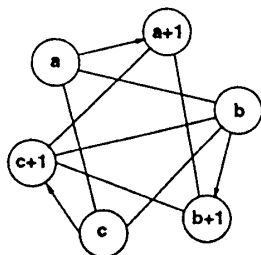


図2: スライドウィンドウを考慮した干渉グラフの例

3.2.1 彩色アルゴリズム

拡張された干渉グラフの彩色では、矢印の取り扱いが問題となる。矢印で繋がっているノードは、レジスタ番号がスライド幅刻みで連続したものとなる。

ここで、問題をノードの彩色のみに絞って考えると、矢印で繋がっているノードはそれぞれ別のレジスタであり、レジスタ番号の連続性を考慮すれば、矢印を通常の干渉の枝とみなして彩色することができると考えられる。その際、問題となるのは図2の例のように、矢印で繋がる変数が複数ある場合に、矢印を考慮しないで彩色するとレジスタ番号の連続性が失われる可能性があることである。

しかし、実際のループプログラムでは、矢印で繋がる複数の変数間には、図2のように必ず干渉があるので、レジスタ番号の連続性は保たれるため、矢印を通常の干渉の枝とみなして彩色し、各色にレジスタ番号を割り付ける場合に番号の連続性を考慮するアルゴリズムを採用する。

3.3 サイクリックインターバルグラフを用いるアルゴリズム

3.3.1 グラフの拡張

サイクリックインターバルグラフとは、図3に示すようなもので、各変数の定義点や使用点・消滅点の情報を明確に表すことができる。従って、スライド命令の前後でレジスタ番号が変化するところがわかっているので、グラフ自体に拡張すべきところはない。

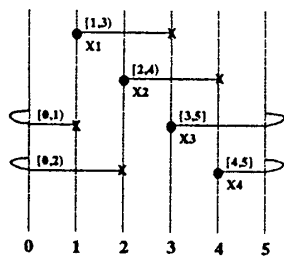


図3: サイクリックインターバルグラフの例

3.3.2 彩色アルゴリズム

ファットカバーアルゴリズムは、ループの複数の繰り返しにまたがる変数に着目してサイクリックインターバルグラフを彩色するアルゴリズムである。スライドウィンドウを用いるオブジェクトコードには、スライド命令が存在し、その前後でレジスタ番号が変化する。従って、スライド命令がループの最後に実行されるようにスケジューリングを行えば、複数のステージ

にまたがる変数を別のレジスタとして彩色できる。彩色は、グラフを左からサーチし、同じ色で塗れるインターバルを探すことを繰り返すことで行う。

4 実験

干渉グラフを用いるアルゴリズムと、サイクリックインターバルグラフを用いるアルゴリズムを比較するための実験を行う。

4.1 実験の方法

2つのアルゴリズムを実装し、さまざまなデータを与えて、それぞれ彩色に要した色の数を比較する。

実験に用いるデータは、テストプログラム自動生成ツールのデータを使用する。テストプログラム自動生成ツールは、ループのプログラムを自動的に生成し、そのプログラムについて命令スケジューリングを行い、変数のライフレンジを出力する。その出力されたライフレンジに対して、レジスタ割り付けを行うものとする。

4.2 実験の結果

テストデータとして、10000個のデータを作成したレジスタ割り付けプログラムに与え、干渉グラフを用いるアルゴリズムとサイクリックインターバルグラフを用いるアルゴリズムで彩色を行った。それぞれのアルゴリズムで彩色に要した色の数を比較したが、両者の差はほとんど見られなかった。

5 まとめと今後の課題

実験の結果から、干渉グラフのアルゴリズムとサイクリックインターバルグラフのアルゴリズムに、ほとんど差は見られなかったが、プログラムとしてはサイクリックインターバルグラフを用いるアルゴリズムの方が短いステップで彩色できることがわかった。

また、レジスタが足りなくなったときには変数の退避・回復を行うことになるが、サイクリックインターバルグラフを用いると変数の生存区間が明確なため、退避・回復の対象となる変数が選びやすくなるという利点がある。

今後の課題は、そのようなレジスタが足りなくなったときの処理も含め、実際のコンパイラにスライドウィンドウを考慮したレジスタ割り付けルーチンを実装することである。

参考文献

- [1] Nakamura.H.,Imori.H.,Nakazawa.K.,Boku.T.,Nakata.I., Yamashita.Y.,Wada.H.,and Inagami.Y. :‘A scalar Architecture for Pseudo Vector Processing based on Slide Window Registers’,ACM Inter. Conf. on Super Computing ‘93 (1993) pp.298-307. 他
- [2] G.J.Chaitin:Register ‘Allocation and Spilling via Graph Coloring’, In Proceedings of the ACM SIGPLAN’82 Symposium on Construction, pp.98-105.SIGPLAN Notices,June 1982. 他
- [3] Laurie J. Hendren,Guang R. Gao,Erik R. Altman,and Chandrika Mukerji: ‘A Register Allocation Framework Based on Hierarchical Cyclic Interval Graphs’, LNCS 641,pp.176-191,1992.
- [4] 筑波大学計算物理学研究センター: 専用並列計算機による「場の物理」の研究, 新プログラムによる研究, 研究進捗状況報告書, 平成6年8月