Superscalarプロセッサのループ最適化と命令レベル並列性について

4M−4

井上淳　白川健治
(株)東芝 総合研究所

1はじめに

VLIW、superscaler等のアーキテクチャでは、1つのMPU内に複数の演算器を持ち、複数個の機械語命令を並列に実行することにより高速化を図っている。このようなアーキテクチャを有効に活用するにはリソースを考慮した個々の命令の並べ換えに加え、プログラム構造に応じた最適化が必須である。

我々はこれまで既存のRISCプロセッサをベースにしたSuperscalarプロセッサを想定して、非数値計算での命令レベル並列性を評価してきた[1]。本稿では、さらに浮動小数点プロセッサを想定してループ実行が支配的な数値計算プログラムに関しての評価を行い、各種ループ最適化方式の適用可能性、命令レベル並列性を評価した。

2マシンモデル

既存のRISCプロセッサの演算器を複数化したものを想定する。そのアーキテクチャは、

- 整数演算器が2個(乗除算はlatencyあり)
- 浮動小数点加算、乗算器が1個(latencyあり)
- 整数ロード命令は2サイクル実行
- 浮動小数点ロード命令は3サイクル実行
- 分岐命令は1サイクル実行
- レジスタは整数、浮動小数点とも32個

とする。各ユニットはパイプライン処理される。ハードウェアによるリソースロッキングは仮定しておらず、コンパイラは各命令のlatencyを考慮して命令を再配置し、リソース衝突のない命令群のみを同時に演算器に供給する。リソース衝突がない場合は、最大4命令(但し整数命令は2個まで、分岐命令は1個のみ)が同時実行できるSuperscalarプロセッサである。

3最適化(並列化)手法

以上のように仮定したSuperscalarプロセッサ用のコンパイラ及び並列化オプティマイザを作成して評価を行った。このオプティマイザは[1]で使用した基本ブロック内の命令列スケジューリングに加え、ループアンローリングを任意の段数で行う機能を持っている。さらにソフトウェアパイプライニング[2]を施すための条件(データ依存、ループ内分岐、脱出、関数コール等)を調べ、各ループ毎にパイプライニング可否を判定する機構を持つ(パイプライニングされたコードはまだ出力していない)。今回の評価では、この機構を用いて各ループのパイプライニング可能性を判定し、パイプライニング可能なものについては仮想的にソフトウェアパイプライニングを施して、ループアンローリングと命令スケジューリングを同時に行った場合との比較を行った。

4評価結果

評価は並列化オプティマイザの生成したコードのうち、ループ内部分の、(実行命令数)/(実行サイクル数)を測定した値(IPC値)をもって行う。ループアンローリングに対する評価は、アンロール段数を0段(アンローリングなし)、4段、8段、16段、32段と変化した場合のIPC値を測定した。また仮想ソフトウェアパイプライニングに対する評価は、パイプライニングのPrologue,Epilogue部分を含めて反復数(最内側ループが1000回反復すると仮定した)を考慮してループ定常実行部分を加重した値を使用する。なおパイプライニング段数は、全て2段とする。

Livermore Kernel(Loop1～12)を対象にして評価を行った結果を表1に示す。値が**のものは、未定なデータ依存、ループ型式、レジスタ数の制限のため、現行のオプティマイザではアンローリング、パイプライニング不可能と判定されたものである。

この結果から以下の考察が得られる。

1)浮動小数点演算を含むループでは、特別な処理を行わない場合、その命令レベル並列度は通常の非数値計算プログラムにおける並列度より小さく、大部分が1.0前後である。

これは、今回使用したマシンモデルにおける浮動小数点演算のlatencyが大きく(加算、乗算とも3サイクル)、前段の演算結果待ちで演算器が有効に使用できないためである。実際、配列アクセスのアドレス計算と通常の浮動小数点演算は大部分が並列化できており、配列計算が支配的なループ(#8)では未処理でも1.8程度の並列度が得られる。

2)ループアンローリングでは、ループ内の演算数、演算種類によっては一定段数で並列度上昇が止まったり、逆に

Loop Optimization for Superscalar Processors and its Instruction Level Parallelism
Atsushi Inoue, Kenji Sirakawa
Research&Development Center,Toshiba Corp.

並列度が低下する場合もある。

ループアンローリングの場合、通常のアクセスパターンのループ(#1、#12)では段数増加に伴い並列度が上昇し、ある点からはレジスタあふれのため逆に並列度が低下する。従って最適なアンロール段数の決定が重要である。一方、データリカージョンを含むループ(#3、#5、#6、#11)やループ帰納変数によるアクセスを含むループ(#2、#4)ではアンロールの効果は小さい。これはアンロールにより基本ブロックが長くなっても、依存データ、帰納変数計算部分の制約条件のため命令移動の可能性がふえないためと考えられる。

3)ソフトウェアパイプライニングは広範囲に適用可能であるが、命令レベル並列性で多数段アンローリングを行ったものに劣る場合もある。

ソフトウェアパイプライニングはアンロールの効果がないデータリカージョンを含むループでも並列化効果を発揮できる。しかし単純なループの場合に、今回の評価では多数段アンロールより劣る場合があった。これは今回のパイプライン段数が小さかった(2段)ため、命令スロットに空きが残ってしまったためであり、本来は各ループに対して最適なパイプライン段数を決定して処理を行うべきである。また#11、#12などのようにループ内演算数が小さくパイプライン段数を大きく取れない場合は、予め2段アンローリングしたものをパイプライニングするなどの処理が必要であろう。また、帰納変数(#2、#4)やループ内テンポラル(#10)の処理方式、ループが大きい場合(#8)の処理方式を確立し、適用範囲を拡げていくことも今後の課題である。

5おわりに

浮動小数点プログラムの場合、演算のlatencyのためループ部分命令レベル並列度は1.0前後である。従って何らかのループ最適化が必須であり、今回はループアンローリングとソフトウェアパイプライニングについて、比較、評価を行った。

ループアンローリングは単純なループに適当な段数を選択して適用すると効果が大きく、パイプライニングより高い並列度が得られることもある。一方、パイプライニングはデータリカージョンを含むループで有効である。

今後はソフトウェアパイプライン部の高度化と、ループ属性に応じた最適化手法の選択方式を検討して、評価を行っていく。

§文献

[1]白川他:非数値計算プログラムにおける命令レベルの並列性について、第41回情報処理全国大会4E-5(1990)

[2]Lam,M:Software Pipelining:An Effective Scheduling Technique for VLIW Machines, Proc. of SIGPLAN Conf. of Prog. Lang. Design and Implementation, ACM, pp.318-328

表1. Livermore kernel(Loop1～12)の処理結果

Loop #	ループアンローリング+命令スケジューリング					パイプライニング
	0段	4段	8段	16段	32段	
1	0.824	1.609	1.917	1.928	1.138	1.748
2	1.154	**	**	**	**	**
3	0.889	1.056	1.167	1.241	1.284	1.998
4	1.000	0.811	0.795	0.786	0.755	**
5	0.913	0.862	0.849	0.842	0.766	1.374
6	1.100	0.757	0.712	0.690	0.667	1.832
7	1.185	1.359	1.156	0.913	0.815	1.777
8	1.842	**	**	**	**	**
9	0.861	1.541	1.576	1.562	1.652	1.887
10	1.000	**	**	**	**	**
11	1.000	0.941	0.966	0.981	0.924	1.599
12	1.143	1.600	2.000	2.364	0.883	1.998