

CHA-MEN 広域命令スケジューラ用プロファイラの実装

米田 淳一† 青木 勲† 岡 大輔† 古川 文人‡ 月川 淳†
 大津 金光† 横田 隆史† 馬場 敬信†
 宇都宮大学工学部情報工学科† 帝京大学ラーニングテクノロジー開発室‡

1 はじめに

我々はVLIW(Very Long Instruction Word)プロセッサシミュレータの共通の枠組みとなるVLIWシミュレーション環境CHA-MENの開発を行った^[1]。

VLIWの性能はスケジューリング方法に依存する。スケジューリングの方法には大きく分けて、局所命令スケジューリングと広域命令スケジューリングとがあり、VLIWの性能をより引き出すためには広域命令スケジューリングが必要である。広域命令スケジューリングの一つにスーパーブロックスケジューリング^[3]がある。スーパーブロックスケジューリングはトレースを決定するためのプロファイル情報を必要とする。我々が実装したCHA-MENスケジューラ^[2]はアセンブリコードを元にスケジューリングを行う。スーパーブロックスケジューリングを行う際には、プロファイル情報を得て、適切な位置にプロファイル情報を挿入する必要がある。また、CHA-MENシミュレータは実行ログを出力する機構を持つので、実行ログを解析することで、プロファイル情報を得ることができる。その際、膨大なログデータからプロファイル情報を人手で得るのは非現実的であり、何らかの自動化ツールで扱う必要がある。本稿では、プロファイル情報を自動で取得し、アセンブリコードの適切な位置に挿入するプロファイラ(以下、CHA-MENプロファイラ)を実装し、スーパーブロックスケジューリングを効率的に行うことができることを示す。

2 VLIWシミュレーション環境CHA-MEN

CHA-MENは、言語処理系、シミュレータ、命令スケジューラから構成されている。図1にCHA-MENの構成図を示す。まずCソースコードをコンパイルすることでアセンブリコードを得る。これを命令スケジューラに入力することで、スケジューリングされたアセンブリコードを得る。それをアセンブラに入力することで、オブジェクトファイルが出力される。このオブジェクトファイルとランタイムライブラリからリンクがイメージファイルを作成する。シミュレータはイメージファイルを入力として、シミュレーションを行い、ログを出力する。

本稿で説明するCHA-MENプロファイラはこのログを解析して、プロファイル情報を得る。スーパーブロックスケジューリングを行う際にCHA-MENスケジューラが必要とするプロファイル情報は、分岐確率情報とループ回数情報である。分岐確率情報とはある分岐命令が何回実行されて、そのうち何回分岐が行われたかを示すものである。ループ回数情報は、ループの実行1回あたりの平均イテレーション数の情報である。プロファイル情

報は図2のようにコード中の基本ブロックの先頭に記述する。分岐確率情報は分岐命令がある基本ブロックに、ループ回数は、ループの先頭となる基本ブロックに、それぞれ記述する。

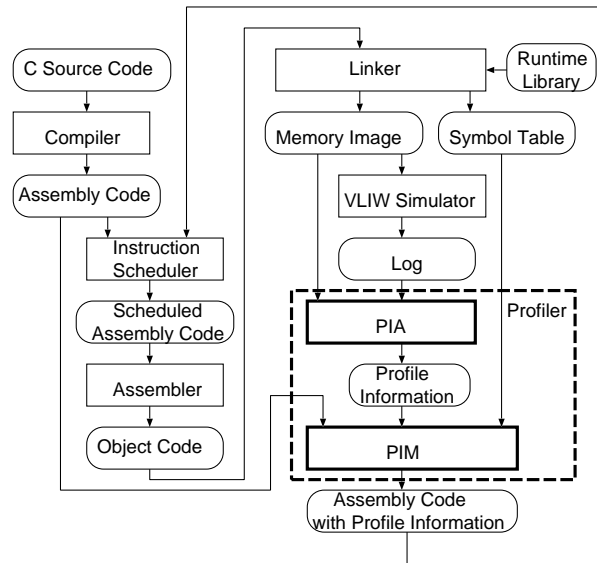


図 1: CHA-MEN 構成図

```

    $L39:
    .prof Number_of_Taken:159/160 Loop_Iterations:160
    sb    $5,0($7)
    addiu $7,$7,0x0001
    
```

図 2: プロファイル情報を入力したアセンブリコードの例

3 CHA-MEN プロファイラの実装

CHA-MENプロファイラの動作は大きく二つに分けられる。第一に、シミュレータの実行ログからプロファイル情報の取得を行うことである。分岐確率情報は、実行ログを解析し、実行された分岐命令の回数と分岐命令の条件成立回数をカウントすることで取得する。ループ回数情報は、アセンブリコードで分岐命令のバックエッジを元にループを特定する。第二に、取得したプロファイル情報をアセンブリコードの適切な位置に挿入することである。これにより、スケジューラはプロファイル情報入りアセンブリだけを読んでスケジューリングを行うことができる。

これらの動作を動作検証や保守性の観点から、二つのモジュールに分割した。一つは実行ログからプロファイル情報を取得するPIA(Profile Information Analyzer)。もう一つはプロファイル情報とアセンブリからプロファイル情報入りアセンブリを出力するPIM(Profile Information Merger)である。

Implementation of a Profiler for CHA-MEN Global Instruction Scheduler
 †Junichi Yoneda, Isao Aoki, Daisuke Oka, Atsushi Tsukikawa, Kanemitsu Ootsu, Takashi Yokota and Takanobu Baba, Department of Information Science, Faculty of Engineering, Utsunomiya University
 ‡ Fumihito Furukawa, Learning Technology Laboratory, Teikyo University

3.1 PIA

PIA はシミュレータが出力する実行ログとイメージファイルを入力とし、プロファイル情報を取得するモジュールである。

PIA はまず、シミュレータ上で実行したイメージファイルを受け取り、分岐命令とループについての情報を得る。分岐命令については分岐命令のアドレスと分岐先のアドレスを、ループについてはループのバックエッジのアドレスとバックエッジの飛び先のアドレスを、それぞれ取得する。その後実行ログを読み込み、プロファイル情報を取得する。PIA はログにある実行された命令のアドレスを読み取り、以下の場合について適切な処理を行うことでプロファイル情報を収集する。

- 実行された命令のアドレスが分岐命令のものか
- 実行された命令のアドレスがループのバックエッジのものか
- 直前に実行された命令が分岐命令やループのバックエッジの場合、その飛び先であるか

分岐確率情報の場合、当該分岐命令の実行された回数を1増やし、その分岐先のアドレスを保持する。さらに次の命令のアドレスが分岐先のアドレスと一致する場合は、当該分岐命令の分岐が実際に行われた回数を1増やす。

ループ回数情報については、バックエッジである場合は、バックエッジの飛び先のアドレスを保持する。さらに次の命令のアドレスが飛び先のアドレスと一致する場合は、通常のイテレーションである。一致しない場合はループを抜けたことを意味するので、イテレーション回数を保存し、次のイテレーションは0から数え直す。バックエッジの飛び先のアドレスである場合、直前がバックエッジならば通常のイテレーションであるため、イテレーション回数を1増やす。一方で、直前がバックエッジでない場合は、ループに突入したことを意味する。

3.2 PIM

PIMはPIAが出力したプロファイル情報、アセンブリコード、シンボル情報を入力し、プロファイル情報入りアセンブリを出力するモジュールである。

シンボル情報とは、アセンブリコードとメモリアドレス上の命令とをPIMが対応づけるために必要なファイルである。アセンブリコードはアセンブラによってオブジェクトファイルに変換された後、リンカによって再配置される。そのため、実行ログが出力するのは実行された命令のアドレスだけであり、アセンブリコード上のどの命令が実行されたかは、アセンブリコードや実行ログを見ただけでは分からない。そこで、シンボル情報を入力とする。シンボル情報は再配置した結果のアドレスとアセンブリコード内にあるラベルとを対応づけており、この情報を参照することで、CHA-MEN プロファイラはアセンブリコードのどの位置にプロファイル情報を挿入するかを決定する。

以下にPIMの動作を示す。まず、シンボル情報を読み込むことで、ラベルとアドレスの対応を付ける。次にアセンブリコードを読み、プロファイル情報を入力できるようにする。最後にプロファイル情報を一一つ読み込み、適切な位置にプロファイル情報を挿入する。

読み込むプロファイル情報が無くなったら、プロファイル情報入りアセンブリをファイルに書き出して、処

理を終了する。このプロファイル情報入りアセンブリをスケジューラに入力することで、スーパーブロックスケジューリングを行う。

4 CHA-MEN プロファイラを用いたスーパーブロックスケジューリングの評価

4.1 評価方法

CHA-MEN プロファイラを用いたスーパーブロックスケジューリングの評価を行う。表1のパラメータで示すプロセッサモデルに合わせてスケジューリングを行った。評価には、SIMCA 用クロスコンパイラを使用し、最適化オプションは、-O2とする。評価対象プログラムとしてNqueen問題(N=13)を使用する。比較対象としてCHA-MEN プロファイラを用いたスーパーブロックスケジューリングの他に、スケジューリングを行わなかった場合と、リストスケジューリングを行った場合のシミュレーションを行う。

表 1: シミュレーションパラメータ

最大同時実行RISC命令数	4
INTユニット数	4
FPユニット数	1
ロードストアユニット数	1
実行ステージでの各演算時間	1サイクル
命令・データキャッシュサイズ	32KB
命令・データキャッシュレイテンシ	1サイクル
キャッシュミスペナルティ	10サイクル

4.2 評価結果

シミュレーションの結果、スケジューリング無しの場合のサイクル数を基準とした場合、リストスケジューリングは1.03倍、スーパーブロックスケジューリングは1.15倍、プログラムを高速化できることが確認できた。

5 おわりに

本稿では、スーパーブロックスケジューリングに必要なプロファイル情報を取得するCHA-MEN プロファイラの実装について述べた。今後の課題として、より多くのプログラムでの評価が挙げられる。

謝辞 本研究は、一部日本学術振興会科学研究費補助金(基盤研究(B)18300014,同(C)16500023,若手研究(B)17700047)、宇都宮大学重点推進研究プロジェクトおよび帝京大学理工学部教育・研究推進特別補助金の援助による。

参考文献

- [1] 月川淳, 古川文人, 青木隆行, 岡大輔, 大津金光, 横田隆史, 馬場敬信, “CHA-MEN: スケジューラ協調開発を支援するVLIWシミュレーション環境”, CPSY, 信学技報, Vol.105, No.453, pp.1-6, (CPSY2005-27), 2005年.
- [2] 岡大輔, 古川文人, 月川淳, 大津金光, 横田隆史, 馬場敬信, “VLIWシミュレーション環境CHA-MENにおける拡張性を重視した広域命令スケジューラの実装”, SACSIS 2006 論文集, 情報処理学会シンポジウムシリーズ Vol.2006, No.5, pp.256-257, 2006年5月.
- [3] 安藤秀樹, “命令レベル並列処理・プロセッサアーキテクチャとコンパイラ”, コロナ社, pp.158-170, 2005年10月.