6G-04

ヘテロジニアスメニーコアプロセッサにおける最適並列処理の決定方式に関する検討

菊池 祐貴 † 大津 金光 † 馬場 敬信 †† 横田 隆史 † 大川 猛 †
†宇都宮大学工学部情報工学科 ††宇都宮大学オプティクス教育センター

1 はじめに

近年，一つのチップ上に複数の CPU コアと GPU を混載した SoC(System-on-Chip) が登場しており，また CPU コアについても省電力などを目的として複数のタイプのものを混載しているものが現れてきている状況で，それらをうまく活用した並列処理技術の開発が必要とされる．活用方法として，プログラムの各部分ごとにプロファイル情報に基づいてそれぞれの並列性を解析し，各並列性に応じて最適なコアアーキテクチャを活用した並列実行を行うシステムを想定とする．そのためのプロファイル情報を取得するプロファイラ [1] について，現在，取得可能である情報に基づいて並列処理方式を決定することの問題点と改善案を述べる．

2 前提とする自動並列化システム

本研究が前提とするヘテロジニアスメニーコアプロセッサ上で自動並列化を行うシステムの概要を述べる．図 1 にシステムの全体構成を示す．

オンラインプロファイラは入力された逐次プログラムとデータセットを実行して動的な情報を取得する．取得したプロファイル情報と，CPU/GPU のコア数等のアーキテクチャパラメータ，動作周波数，消費電力，発熱等の制約条件を入力として，オフラインアナライザはコード領域ごとに解析を行い，最適な並列処理方式を決定する．そして，オフラインアナライザの判定結果を元にプログラムトランスレーターが各コード領域ごとに並列コードへの変換を行う．変換された並列化コードを既存のコンパイラを用いてコンパイルを行い，各コード領域ごとにヘテロジニアスメニーコアプロセッサで実行を行う．

本システムでは，各プログラムコード部分ごとに以下の処理方式のいずれかが最適であるかを決定し，コード変換を行うかを判定する．

- マルチコア CPU によるマルチスレッド実行
- GPU による大規模データ並列処理
- SIMD 演算命令による小規模データ並列実行
- シングルコアによる逐次処理実行

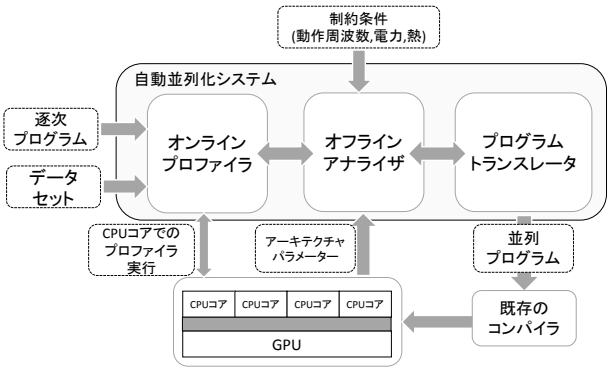

図 1. ヘテロジニアスメニーコアプロセッサ向け自動並列化システム

3 最適並列処理の決定方式とその課題

我々が開発しているプロファイラ Pathgrind は，動的バイナリ変換ツールである Valgrind[2] のプラグインツールという形で実現されている．Valgrind は基本ブロック単位で仮想マシン上で実行が行われるため，ループのプロファイル情報を取得するためには，実行される基本ブロックをパスとして保持する必要がある．パスの終端は関数のコール，リターン，後方分岐である．そのため，外側のループは最内ループの終端の後方分岐でパスが切断されるため検出することができない．取得可能な情報は，ループ一回あたりのイテレーション数，総イテレーション数，ループに含まれるパスの本数，パスごとの実行命令数，ループ全体の総実行命令数と同パスを並列実行した際のデータ依存の有無である．

並列化対象プログラム内のあるループの並列処理方式を決定するにあたり，ループの個々の実行パス間に依存関係があるかどうかを判定する必要がある．依存関係がないと判断された場合は，並列化可能であると判定し，並列処理方式を決定する．依存関係があると判断されたループに対しては逐次実行が最適な処理方式と決定する．最適な並列処理方式を，イテレーション数からデータ並列処理の規模を決定する．パスが複数存在する場合はマルチコア CPU によるマルチスレッド並列処理を最適な並列処理方式とする．

ループの並列化を検討するときには，様々な並列化方針がある．様々な視点から並列化を検討することは並列化対象ループに対してより適した並列処理方式を検討できると考えられる．例えば，2 重ループの場合に，外側のループに対して並列化を検討する方が適している場合と，最内ループに対して並列化を検討する

Consideration on Determination of Appropriate Scheme of Parallel Processing for Heterogeneous Many-core Processor System

†Yuki Kikuchi, †Kanemitsu Ootsu, ††Takanobu Baba, †Takashi Yokota and †Takeshi Ookawa
Department of Information Science, Faculty of Engineering, Utsunomiya University (†)
Center for Optical Research and Education, Utsunomiya University (††)

方が適している場合がある．しかし，現状のプロファイラでは最内ループのみ情報を取得するため，外側のループに対して並列化を検討することができない．そこで，最内ループだけではなく，外側のループに対してプロファイル情報を取得できるようにし，並列化対象ループに対してより適した並列化処理方式を決定できるようにする必要がある．

4 解析範囲の拡大

前節で述べた通り，より適した並列処理方式を決定するために解析範囲の拡大が必要である．現状のプロファイラを拡張して最内ループの外側のループに対してプロファイラ情報を取得できるようにする方式を述べる．基本的には最内ループの解析を行ってから，その外側のループが存在する場合はそれも解析の対象とするという形でループ全体を解析していく．その際，内側の解析済みのループを一つのマクロブロックとして扱う．内側のループの前後でパスが分かれているため，これらをつないで一本のパスとして扱う．マクロブロックに入る前のパスとマクロブロックを出た後のパスを繋げる．

通常の基本ブロックは命令数についての情報を持つがループを置き換えたマクロブロックではループ中のパスに含まれる命令数とループ一回当たりのパスの平均実行回数の積をブロックの命令数として扱う．マクロブロックは元々のループを一つのブロックとして置き換えたものであり，基本ブロックと同様に入力データと出力データ内部で使用したレジスタやメモリ情報を含むこれらの情報を用いて，外側のループでのイテレーション間の依存関係を解析することができる．

5 具体例

実際に MiBench suite[3] の blowfish(ecb) を例にして多重ループの解析を行う．図 2(a) が対象多重ループのフロー図である．blowfish (ecb) は入力ファイルとして平文を用意し，入力したキーに応じて，出力ファイルとして暗号文が生成されるというプログラムである．対象多重ループは 2 つの最内ループを含み，1 つ目のループで input に暗号化を行うデータを格納する．2 つ目のループで暗号化したデータの output を出力ファイルに出力する．それぞれの最内ループの一回あたりのイテレーション数は 40 であるが，総イテレーション数は 32 万を超える．つまり，外側のループのイテレーション数が大きいことがわかる．これは，従来の我々のプロファイラは最内ループのみ解析の対象としているため，このループに対して有効な並列処理を判定することができない．そこで，解析範囲の拡大を行い，外側のループを解析する．定義後のフロー図を図 2(b) に示す．最内ループをマクロブロックとして定義することにより単体ループとして対象ループを解析することができる．解析する際にマクロブロック内の情報を解析結果に反映させ，対象ループの解析を行う．これらを実現することにより，これまで検討する

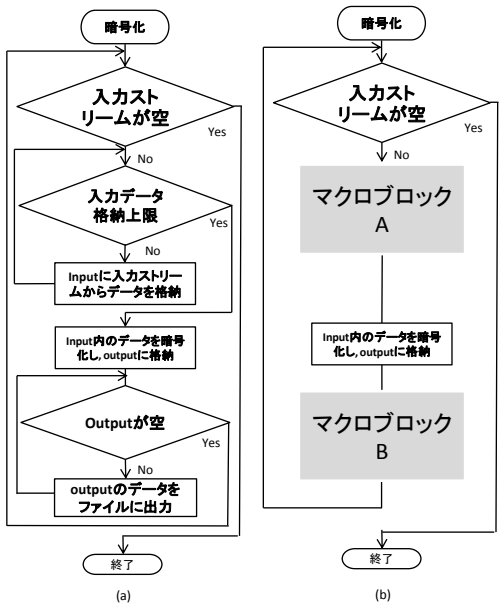

図 2. (a)blowfish のフロー図 (b)blowfish のフロー図 (定義後)

ことができなかった並列化を検討することができるようになり，より適した並列化処理方式を検討することができると考えられる．

6 おわりに

本稿では，プログラムのそれぞれのループに対して最適並列処理方式を決定する手順と現状の決定方式に関する問題点を述べ，最内ループだけではなく外側のループも考慮に入れて並列処理方式を決定することが必要であることを述べた．今後はそれが可能なプロファイルを開発する．

謝辞

本研究は一部 JSPS 科研費 24500055，25330055，15K00068 の助成による．

参考文献

[1] 大島 一樹, 大津 金光, 馬場 敬信, 大川 猛, 横田 隆史: “プログラム実行パス間のデータ依存を解析するためにパスプロファイラの実現”，信学技報，Vol.114，No.155，pp.203-208，2014.

[2] Nicholas Nethercote and Julian Seward: “Valgrind: A Framework for Heavyweight Dynamic Binary Instrumentation”, Proc. of ACM SIGPLAN 2007 Conference on Programing Language Design and Implementation (PLDI 2007), pp.89-100, 2007.

[3] M.R.G, J.S.R, D.E, T.M.A, T.M, and R.B.B: “Mibench: A Free, Commercially Representative Embedded Benchmark Suite”, Proc. of IEEE 4th Annual Workshop on Workload Characterzation, pp.3-14, 2001.