Technical Reports on Information and
Computer Science from Kochi
Vol. 8 (2016), No. 5

オブジェクトソフトウェアのプロファイリング技術の確立

Profiling Methodology of Object Software

岩河　秀知　　村岡 道明
Hidetomo Iwagawa　　Michiaki Muraoka
高知大学大学院 理学専攻（情報科学分野）

1. まえがき

現在，ソフトウェアの最適化を行うため，C 言語で記述されたソフトウェアのプロファイリング技術が開発[1]されているが，オブジェクトコードしかないプログラムについては，その構造や性能の解析技術は未確立である．本手法ではオブジェクトコードの構造および基本ブロックやサブルーチンの動作回数や実行時間などを解析し，性能に関するボトルネックの検出を行う．本技術の確立によりソフトウェアの再利用性の向上を図ることが可能であることを示す

2. ソフトウェアオブジェクトの解析手法

本組込みコード解析手法はターゲットプロセッサのオブジェクトコードを基本ブロック単位に分割し，構造や性能を解析する手法である．解析機能としてオブジェクトコードの逆アセンブル変換，制御構造の生成，基本ブロック・サブルーチン単位でのプロファイリングを行う．解析部の構成は静的解析部と動的解析部から構成される．

2.1 静的解析機能

動的解析部では，オブジェクトコードを基本ブロックに分割，フローチャートの生成および静的解析項目のプロファイリングを行う．入力から出力までの手順を図 1 に示す．入力は命令オブジェクトを 16 進ダンプしたものである．

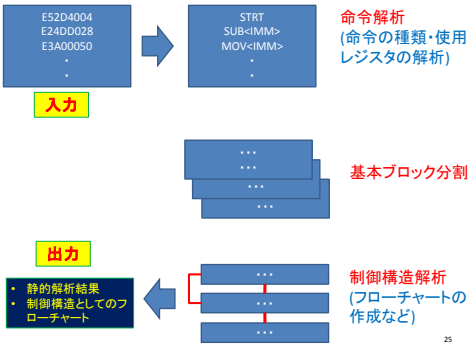

図 1　静的解析手順

2.2 動的解析機能

静的解析部では解析を行うために命令セットシミュレータ(ISS)を作成[2][3]し，命令の実行と同時に動的解析を行う．入力から出力までの手順を図 2 に示す．入力は静的解析で命令解析を行った命令を 1 命令ずつ ISS に読み込む．

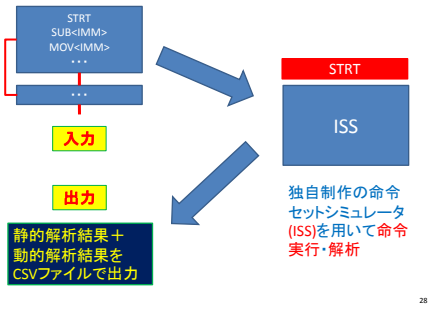

図 2　動的解析手順

3. 解析評価結果

本組込みコード解析手法を用いて命令数が 100 以下の基本的なプログラムの解析を行った．解析に用いた基本的な C プログラムの一覧を以下に示す．

- データ 1：クイックソート (QSort)
- データ 2：バブルソート (BSort)
- データ 3：平均値算出プログラム (Average)
- データ 4：ハノイの塔 (Hanoi)
- データ 5：行列計算プログラム (Matrix)

また，これに加え実用的なプログラムとして IEC7816 ミドルウェアに本手法を適用した．全体のサイクル数と比較して，各基本ブロック当たりの実行サイクル数の占有率を示す帯グラフを図 3 に占有率の大きいものから順に示す．基本ブロックの ID (#)，サイクル数，占有率を示す．

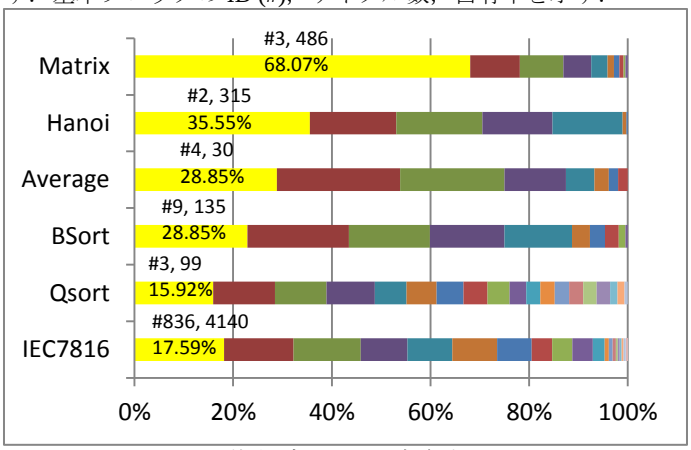

図 6　基本ブロックの占有率

解析の結果，Matrix($O(n^3)$)，Hanoi ($O(2^n)$)とオーダーの大きなものは 1 つの基本ブロックが全体の高い占有率を占めていることが確認でき，ボトルネックの検出や性能解析に効果が期待できる．実用的なプログラムへの適用では，IEC7816 ミドルウェアの一部の基本ブロックで評価したところ，大きなボトルネック部分はないという結果となった．

4. まとめ・今後の課題

オブジェクトコードからプログラム構造を解析し，プロファイリングを行う解析手法を提案した．本手法を C プログラムに適用したところ，計算量が大きいほどボトルネックとなる基本ブロックの占有率が高くなることが確認でき，性能解析とボトルネック検出に効果が期待される結果となった．今後は割り込み処理対応，グループ化をしての解析，多数の大規模ソフトウェアへの適用，ターゲットプロセッサを複数サポートし汎用性を高める必要がある．

参考文献

1) 荒木大, “ELEGANT の SpecC シミュレーションと設計詳細化について”, 第 3 回先端宇宙情報技術ワークショップ前刷講演集, pp.1-13, (2007-10)
2) “ARM7TDMI(Rev4)テクニカルリファレンスマニュアル”, (2001-4), ARM Limited
3) “RealView Compilation Tools バージョン 4.0 アセンブラガイド”, (2008-9), ARM Limited