

バイナリコードの動的最適化を実現する 実行時ハードウェア解析システム

請園 智玲 (北陸先端大)・田中 清史 (北陸先端大)

1 はじめに

近年、ソフトウェア開発手法は多様化し、ダイナミックリンクライブラリやダイナミッククラスローディング技術、また JAVA JIT コンパイラのような動的コード生成技術が一般的に使われるようになってきた。これら技術は、実行時にバイナリコードが確定するため、コンパイル時に行われる静的コード最適化技術ではプログラム全体にスコープを当てた最適化を施すことができない。また、静的最適化では、対象プログラムの振る舞いを完全に解析できないため、実行状況予測を用いた最適化を施すことが難しい。今までの研究では、実行時にコードに最適化処理を施す手法として、インタプリタや VM を使った動的最適化技術 [1][2]、また、一度実行した結果をフィードバックして再度、最適化を施すプロファイルベース最適化技術が研究されてきた [3]。本研究では、動的最適化を実現するハードウェアとオペレーティングシステム (OS) サポートによる実行時ハードウェア解析システムを提案し、従来の研究より高速かつ高精度な動的最適化を行う実行環境の構築を目的とする。

2 実行時ハードウェア解析システム

提案する動的最適化機構は OS の一部分としてトラップハンドラで実装される。システムはターゲットコードを実行中にハードウェア/ソフトウェアで主記憶に実行状態ログを残し、最適化ハンドラがトラップで駆動された後、ログを検査し、必要な最適化をターゲットプログラムに施す。

2.1 Hardware Logger

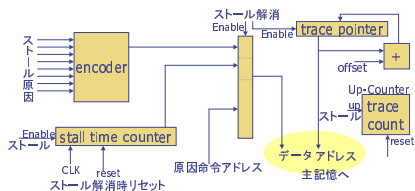


図 1: 実行状態ログを主記憶に残すハードウェア

実行状態ログを主記憶に残すハードウェアを図 1 に示す。このハードウェアは命令実行パイプラインがストールした場合に駆動され、ストールの原因となった命令へのアドレスとストールサイクル数、ストール原因コードを trace pointer レジスタが指し示す主記憶の位置へ書き出す。このハードウェアにより、最適化ルーチンは命令実行の妨げとなった原因とその発生頻度、また、原因がどの程度実行に影響を与えたかを知ることが可能となる。

2.2 Software Logger

ソフトウェアで実行状態ログを主記憶に残すために、ターゲットプログラムの実行を中断しログを残すためのソフトウェアを駆動する。この機能を実現するためのハードウェアの一部を図 2 に示す。本ハードウェアは特定の命令に対して指定動作を行った場合に指定トラップハンドラを駆動

する。例えば、図中の最初のエントリは BNE 命令が実行されて taken だった場合にトラップを駆動する。Software Logger は Hardware Logger より柔軟な振る舞いが可能であり、複雑なデータ構造のログを収集したり、トレースの抽出やループ数解析に利用する場合に効率的である。

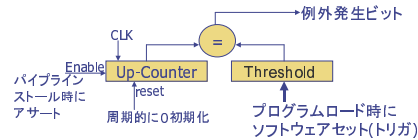


図 2: ログハンドラを駆動するためのハードウェア

2.3 Optimizer Trap

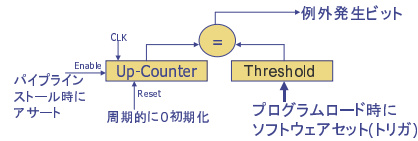


図 3: 最適化ルーチンを駆動するためのハードウェア

システムはターゲットプログラムを実行中、最適化が必要な場所を見つけ、最適化ルーチンを駆動する。このためのハードウェアを図 3 に示す。ハードウェアは命令実行パイプラインストール時にストールしたクロックサイクル数をカウントアップするアップカウンタを保持している。このカウンタ値が OS によって設定した閾値レジスタの値と等しくなった場合にトラップが発生し、最適化ルーチンに制御が移る。また、カウンタ値は OS によって周期的にリセットされ同時に Hardware Logger の trace count と trace pointer も初期化される。本システムではカウンタ値を CPU ストレスと位置付け、区間 CPU ストレスが閾値を越える個所を見つけるハードウェアとして Optimizer Trap を提供する。

3 最後に

本稿において動的最適をサポートするハードウェアと OS の振る舞いを示した。本システムはバイナリコード実行中に最適化のための情報収集と最適化処理を施す仕組みであり、多量の実行時情報を最適化ルーチンに提供できることから、多様な最適化アルゴリズムに適応可能である。

参考文献

- [1] Dynamo: A Transparent Dynamic Optimization System. SIGPLAN Conference on Programming Language Design and Implementation, pages 1–12, 2000. Vasanth Bala, Evelyn Duesterwald, Sanjeev Banerjia Hewlett-Packard Labs
- [2] Design, Implementation and Evaluation of Adaptive Recompilation with On-Stack Replacement. Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization, pages 241–252, Stephen J. Fink and Feng Qian,
- [3] Software Trace Cache. IEEE TRANSACTION ON COMPUTERS VOL.54 NO. 1 JANUARY 2005, Alex Ramirez, Josep-L. Larriba-Pey, Carlos Navarro, Josep Torrellas, Mateo Valero