

性能評価のための命令トレーサの開発

5 F - 7

池谷 敬之 吉澤 康文

東京農工大学大学院 工学研究科

1. はじめに

近年、計算機の資源の増加に伴い、オペレーティングシステム(以下、OS)の果たす役割は大きくなっている。OSの役割は、アプリケーションの資源要求に応じてシステム資源を管理し、システムを効率的に稼働させることである。近年見られるOSは、求められる機能の多様化により巨大化する傾向があり、本質的な動作部分はブラックボックス化している。そのため、OSも含めた計算機全体の処理を把握することは非常に困難になっている。

また、OSはアプリケーションやシステム資源との密接な関係によって成り立っているため、計算機システムの性能評価を行うことは非常に困難であると一般的にいわれている。また、OSの制御方式のアルゴリズムは、OSの性能に深く関わってくる。そのため通常は、設計した後には必ず性能評価を行い、その性能の確認を行う。性能評価を行う場合はその評価を行うための測定を行わなければならない。

本研究では計算機の性能評価を行うための情報を収集する命令トレーサの設計と実装について述べる。

2. 命令トレーサの目的

命令トレーサとは、評価対象の計算機の動作状況に関する情報を集め、それを後に解析データとして使用できる形式で記録するものである。実行に関するすべての情報を記録することにより、その結果をもとにした評価、デバッグ、シミュレー

トを行うことができる。

命令トレーサの動作を考えるにあたって、トレーサの介入によるプローブ効果問題がある。プローブ効果とは、評価対象に情報取得のために割り込むことで、評価対象に突動とは違う影響を与えてしまうことである。これをなくすことは不可能であるが、計測のために生じるオーバーヘッドを最小にし、評価対象の実動の状況にできる限り近い状態のデータを採取する。

3. 命令トレーサの対象

また、本命令トレーサはシステム開発者やプログラマを使用者と考えて開発を行った。

また、CPUはIntel系PCを対象にし、Intelの提供するデバッグ割込みを利用している。OSは最近では、一般的なフリーのUNIXとして普及しているLinuxを使用し、トレーサはカーネル内に組み込む。トレースの対象としては、ユーザレベルのプログラムだけでなく、OSにおける処理も含む。ユーザレベルのみのトレース、カーネルレベルのみのトレースなども行うことができる。

4. トレーサの設計

4. 1. トレーサの構成

本命令トレーサは、1命令ごとにトレース対象プログラムに割り込み、評価対象の実行に関する情報を取得する。トレースを開始する手順としては、まずトレーサを起動し、開始と停止のポイントを利用者が設定する。また、場合によっては取得する必要のない情報についても起動時に設定することができる。

実質的なトレースを行う区間は、トレース開始のポイントの命令を実行しようとした時点からトレースを開始し、停止に指定したポイントまたは、トレー

Development of Instruction Tracer for Performance Evaluation

Hiroyuki IKEYA, Yasufumi YOSHIZAWA

Faculty of Engineering, Tokyo University of Agriculture and technology

サ終了を行うか、起動時に設定した終了条件になったときに停止する。

4. 2. トレーサの取得する情報

トレーサが取得する情報は、実際に実行する命令とその命令実行に関する情報、割込みに関する情報などである。これらは、ファイルに記録し後の解析に使用する。のちに解析を行うときには、トレース結果ファイル解析プログラムにより、解析する。主な取得する情報は、命令コード、プログラムカウンタ、関数名、レジスタ情報、割込み情報などで、後にトレース結果から実行状態を再現することができる。

4. 3. トレーサの起動と終了

トレースの開始、終了は、プログラムで使用されている関数名によって指定する。これにより、トレースの開始、停止を行うポイントの指定を容易に行うことができる。また関数名も記録するので、膨大な量になったトレース結果がどの部分のものであるかを簡単に見分けることができる。

5. トレーサによる評価

本トレーサによって取得した情報から例えば次のような評価を行うことができる。各命令の細かな情報を取得できるのでミクロな評価を行うことができる。

- ・命令の出現頻度の調査
- ・おおよそのダイナミックステップ数
- ・関数ごとの命令ステップ数
- ・ボトルネックとなる部分の把握

例えば、カーネル内に組み込んだプログラムなどでは、ダイナミックステップ数や実行される命令数などを調べることが大切になってくる。そして、関数名ごとにトレース結果を見ることができるので、その結果からどの関数がボトルネックとなっているかを把握することにも使うことができる。

また、メモリを参照している命令ではアドレスも記

録しているのでメモリ参照の局所性といったものや、アドレッシングモードの使用頻度といったコンパイラの性能についての評価も行うことができる。

もちろん、評価を行うだけでなく、デバッグや動作確認、目標としている性能を達成することができたかどうかを調査することにも使うことができる。

6. 実装

各命令の情報は、基本的にはプロセッサの提供するシングルステップトラップで取得している。また、カーネル内で実行される命令の情報を取得するために、本来の OS で使用される割込みディスクリプタテーブル(IDT)の他に、トレーサ用 IDT を用意する。割込みが発生した場合に、まずトレーサ用 IDT に制御が移り、カーネル内のトレースの準備を行う。そして、トレースの準備を行った後で、本来の IDT に制御を移行する。これにより本来の IDT における命令についてもトレースすることができる。トレーサ自身のトレースは行わない。

また、トレーサの開始、終了は関数名をコンパイラの実出力したテーブルを元にアドレスを割り出し、そのアドレスに対するブレイクポイントによって行う。

7. 終わりに

本報告では、計算機の性能評価を行うための命令トレーサの設計と実装について述べた。今後は、本報告で述べた命令トレーサを完全実装し、それを用いた評価を行っていく予定である。

謝辞

本研究は、文部省科学技術研究補助金：基盤研究 C-2-10680338 の成果である。

参考文献

- [1] 吉澤康文 他、計算機システム性能解析の実際、情報処理学会
- [2] Jonathan B. Rosenberg、How Debuggers Work、Wiley Computer Publishing