

アドレ스트レース駆動シミュレータの開発

2N-7

武内 和昭 加藤 宣弘 関戸 一紀

株式会社 東芝 情報・通信システム技術研究所

1.はじめに

計算機の開発段階において、キャッシュやバスなどの動作を考慮に入れた実質的なプロセッサ性能を予測するために、シミュレーション技術が用いられる。そして、性能予測の精度を上げるためには、ワークロードとモデルをいかに正確にするかが課題となる。本稿では、このような課題を解決する手法として、アドレストレース駆動シミュレーション技術を提案する。

本稿では、従来の手法の問題点と今回提案する方法による解決の全体像について述べ、次にアドレストレースの収集方法について説明する。そして、シミュレーション・モデルの構成について詳細に述べる。

2.従来の問題点とその解決法

従来、アドレストレースを用いた予測では、既存計算機で収集したアドレストレースをそのまま用いていたので、プロセッサ数やメモリなどの異なる構成に対しては適応できなかった。また、よく用いられるアクセスパターンを確率的に生成する手法では、構成に対する柔軟性はあるが、[1]で述べられるようなアプリケーションの特性を反映させることが困難であった。

今回、図1で示す構成を取ることで、先の問題を解決する。開発途上の計算機でアドレストレースを収集できないため、まず既存の計算機システムにおいてアドレストレースを採取する。これにより、アプリケーションの動作を反映したワークロードを採取することができるようになる。採取したアドレ

ストレースを、シミュレータに適応することにより、開発途上の計算機のキャッシュミス率、メモリアクセス時間等の性能予測値を得る。アドレストレースを採取した計算機システムが開発途上計算機システムと異なる場合でも、シミュレータが開発途上の計算機のハードウェアの構成やOSの動作をシミュレーションするため、プロセッサやメモリの異なる構成に対して対応ができるようになった。

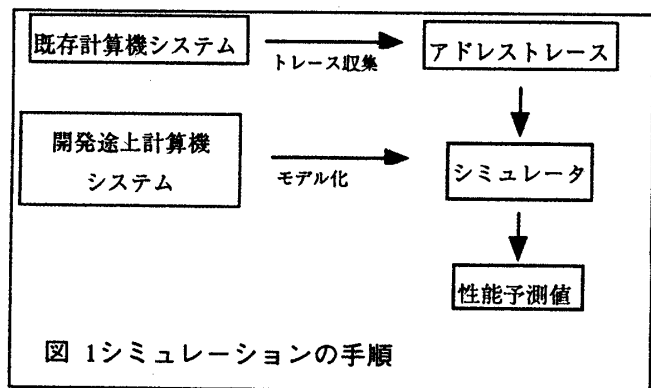


図1シミュレーションの手順

3.アドレストレースの収集

アドレストレースの収集には、Winsconsin大学のJ.R.Larusが作成したアドレストレース収集ツールであるQPT[2]を使用した。

QPTは実行形式のファイルから、実行コードを解析し、必要な箇所にトレースコードを挿入した新しい実行形式のファイルを作成する。トレースコードは命令がアクセスしたメモリのアドレスや情報などプログラムの動作を知るのに必要なコードを記憶領域に書き出す命令である。新しくできた実行ファイルを実行することにより、アドレストレースを収集することができる。

Development of Address Trace Driven Simulator

Kazuaki Takeuchi, Nobuhiro Kato, Kazunori Sekido

Information & Communication Systems Laboratory, TOSHIBA Corporation

2-9 Suehirocho, Ohme, Tokyo, 182, Japan

4. シミュレータの構成

シミュレーションの精度を向上させるために、シミュレータを以下の点にポイントをおいてモデル化を行った。

1. キャッシュ・メモリの状態変化。キャッシュメモリにヒットするか否かにより、メモリアクセスの性能が大きく異なる。そこで、キャッシュ・メモリの動作を忠実にシミュレーションするモデルを作成する。

2. OSのプロセススケジューリング機能。プロセスがどの順序で実行されるかにより、キャッシュメモリの状態がキャッシュメモリの状態、メモリアクセスの状況が異なる。そこで、OSのプロセススケジューリング機能を忠実にシミュレーションするモデルを作成する。

シミュレータの構成を図2に示す。この構成にし

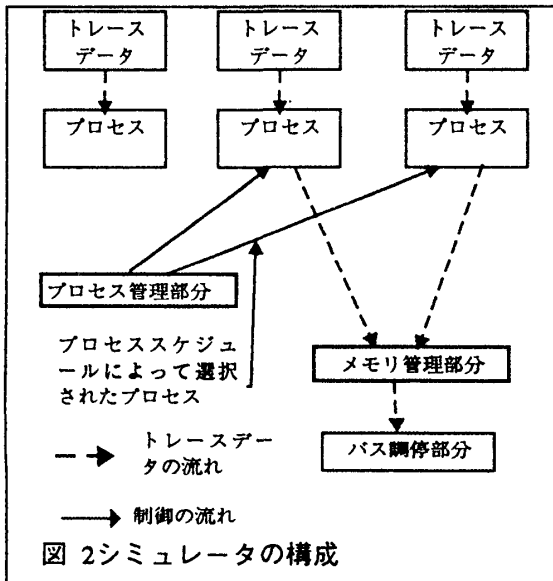


図2 シミュレータの構成

たがって、シミュレータの各部分を説明する。

4.1. メモリ管理部分

この部分では以下の処理を行っている。

1. 仮想アドレスから物理アドレスへの変換。QPTでは、アドレステーブは仮想アドレスで採取される。メモリのアクセスの様子を知るために、仮想アドレスから物理アドレスに変換する必要がある。

2. 対象のメモリアドレスの存在するメモリの決定。メモリアクセス時間は、アクセスの

対象となるメモリにより異なるためである。

3. ページフォルト回数の計測。プロセススケジューリングのパラメータであるページフォルトの回数を計測する。

4.2. プロセス管理部分

この部分では以下の処理を行っている。

1. アドレステーブとプロセスの対応付け。

2. プロセスの優先度を計算。プロセスの実行時間、ページフォルト回数などを用いて計算する。

3. プロセススケジューリングの処理。プロセスの優先度から、実行できるプロセスを選択し、CPUに割り当てる。

4.3. バス調停部分

メモリアクセスがバスを介して行われるときに、バスの調停をシミュレートする部分である。複数のCPUがバスを共有する密結合の計算機では、バスの使用状況によって、CPUのバス要求が待たされることがあり、これにより実質的なプロセッサ速度が変化するためである。

5. おわりに

開発途上の計算機の性能評価においてシミュレーションを用いる場合、シミュレーションのモデルやワークロードの精度が問題となる。本稿では、ワークロードにアドレステーブを用るとともに、シミュレータではキャッシュやバスのハードウェア動作とOSの機能を詳細にモデル化することで対応した。

現在は、より忠実なシミュレーションモデルを作成中で、対象の計算機を使った検証を予定している。

参考文献

1. A. Maynard et al "Contrasting Characteristics and Cache Performance of Technical and Multi-User Commercial Workloads", ASPLOS VI-10, 1994
2. J.R. Larus, "Efficient Program Tracing", IEEE COMPUTER, May, 1993