

トレース駆動シミュレーションによる OS の性能評価

7M-5

堀内谷州子, 茂木 久

(株) 東芝 情報・通信システム技術研究所

1 はじめに

我々は、ワークロードとしてアドレストレース(プログラムの実行した命令と参照したデータアドレスの履歴)を使用したトレース駆動シミュレーションによる、計算機システムの性能評価を行っている。

アドレストレースの収集には QPT(注)[1] を用いているが、この QPT はユーザの実行可能ファイルにしか適用できないという問題点があった。OLTP のベンチマーク処理では、1 つのトランザクションの実行時間の約半分がカーネルモードであり、OS の処理部分でのアドレストレースも必要である。

本稿では、QPT の変更とカーネル内部の改造を行い、カーネルモードでのトレース収集方法を示す。

2 性能評価手法

図 1 に、我々のアドレストレースを使用した計算機システムの性能予測手法を示す。アドレストレースを入力とするキャッシュシミュレータと、その結果であるバス・トランザクションを入力とするバスシミュレータから成る。

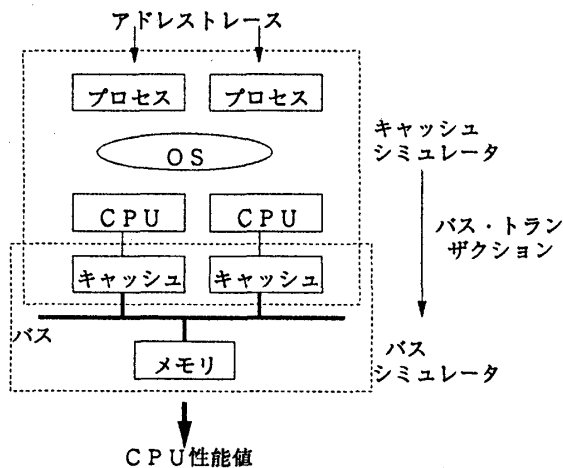


図 1: 性能評価手法

キャッシュシミュレータ [2]

OS のスケジューリングに従ってプロセスを CPU

Evaluation of OS by Trace Driven Simulation
Yasuko HORIUCHI, Hisashi MOGI
Information & Communication Systems
Laboratory TOSHIBA Corp.

注) This study made use of the QP/QPT Performance Tools developed by James R.Larus at the University of Wisconsin-Madison

に割り当て、メモリへのアクセスを含むキャッシュの動作をシミュレートする。また、仮想アドレスで収集したアドレストレースを物理アドレスに変換する。プロセスごとにキャッシュ・ヒット率を予測し、バス・トランザクションのトレースデータが生成される。

バスシミュレータ

バス・トランザクションのトレースデータを入力として、バスの動作をシミュレートし、キャッシュ・ミス・ペナルティや、プロセッサ性能を求めることができる。

このシミュレーションでは、プロセスごとのユーザモード/カーネルモード両方のアドレストレースを使用するため、QPT とカーネルを変更する必要がある。次節以降で、これらの変更について詳述する。

3 QPT の改造

QPT は、ユーザの実行可能ファイルを書き換え、トレース収集コードを埋め込んだ新しい実行可能ファイルを作成する。QPT を OS に対して適用するには、ユーザプログラムと異なり、以下の問題点がある。

- スタックやレジスタの使用やメモリアクセスに制限があり、トレース収集コードを埋め込むことができない部分がある。
- 動的ロードされる部分が存在し、実行時になるまでアドレスが確定しない場合がある。

そこで、QPT を再配置可能なオブジェクトファイルに対しても適用可能にすることで、以下のように上記の問題を解決できた。

- QPT をかけるモジュールを選択できるので、上記で述べたような特殊な制限のある部分にはトレース収集を行わないようにできる。
- オブジェクトファイル内の相対アドレスとしてアドレスを取り扱い、実行時のロードされたアドレスを使って実際のアドレスを生成するようにする。

4 カーネルの改造

前節で述べた改造版 QPT を使用すると、トレースデータは、カーネルモードではカーネル空間に、ユーザモードではユーザ空間に蓄積される。この節では、我々の性能評価手法で必要とする、ユーザモード/カーネルモード両方でのアドレストレースをプロセスごとに収集するためのカーネルの改造について詳述する。

4.1 トレースバッファの管理方法

図2は、カーネル空間、ユーザ空間それぞれのトレースバッファとその管理方法を図示したものである。

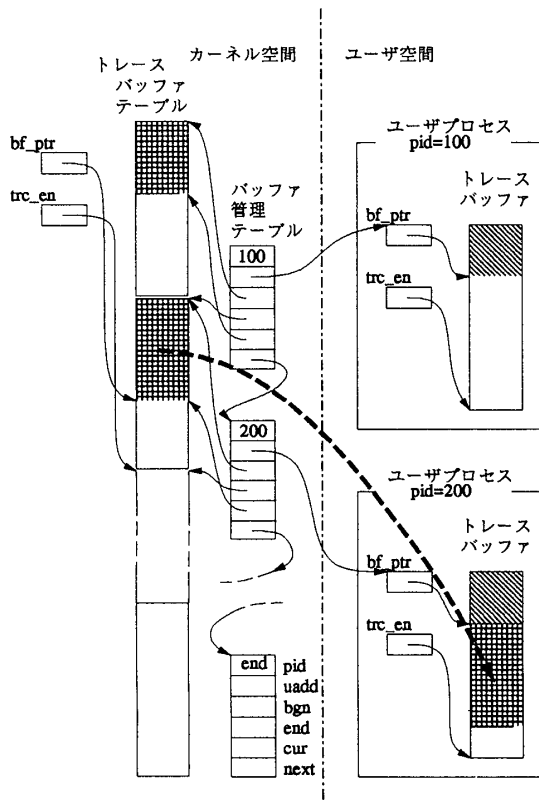


図2: トレースバッファの管理方法

ユーザプロセスにおいては、プロセス内でトレースバッファを持ち、トレースデータはポインタ変数によって示される位置に書き込まれる。バッファが一杯になれば、トレースデータをファイルに書き出し、ポインタをリセットする。

カーネルにおいても、トレースデータはポインタ変数が示す位置に書き込まれる。カーネル内では様々なプロセスが並行に処理されるため、このままではトレースデータが混在し、プロセスごとにトレースデータを抽出できない。そこで、次のデータ構造を用意して、プロセスごとのトレースデータを格納する。
トレースバッファテーブル

プロセスごとのトレースバッファの集まり
バッファ管理テーブル

プロセスごとに、プロセスID、トレースバッファの先頭、末尾、現在位置、ユーザプロセス内のトレースバッファの現在位置を指すポインタ変数のアドレスを格納したリスト

ユーザプロセスを実行すると、トレースデータは、ユーザモードではプロセス内のトレースバッファ、カーネルモードではバッファ管理テーブルで示されるカーネル内のトレースバッファに蓄積される。カーネル空

間のトレースデータはシステムコールのリターン時にユーザモードに移す。

4.2 システムコールハンドラの改造

カーネルモードのトレースデータをプロセス内に移すため、次のようにシステムコールハンドラを改造する。

1. システムコールの処理前に、そのプロセスが使用するトレースバッファの先頭からトレースを書き込むように設定する。
2. システムコールの処理後に、システムコール処理中に書き込まれたトレースデータを、該当するプロセス内のトレースバッファにコピーし、プロセス内のトレースバッファのポインタ変数をデータ分だけ進める。

以上詳述した通り、プロセスごとに、ユーザモード/カーネルモードのトレースデータをまとめて収集できる。

4.3 コンテキストスイッチの改造

カーネル内では、プロセスごとにトレースバッファを用意するため、実行するプロセスの切り替え時に、使用するトレースバッファを変更する必要がある。QPTによって挿入されたトレース収集コードでは、トレースバッファへのポインタ変数と、トレースバッファの末尾位置を示すポインタしか参照しない。よって、コンテキストスイッチの処理内で、次のようにバッファを切り替える。

1. それまで実行していたプロセスのバッファ管理テーブルに、トレースバッファの現在位置を保存する。
2. リストをたどり、次に処理するプロセスのバッファ管理テーブルのデータを取り出す。
3. 次に処理するプロセスのトレースバッファの現在位置と末尾の位置を、この後書き込むトレースバッファとしてポインタ変数に代入する。

5 おわりに

本稿では、アドレストレースを用いた性能評価手法について述べ、カーネルモードでアドレストレースをとるための、QPTの変更とカーネル内部の改造方法について詳述した。

今後は、本方法を用いて、計算機システムの評価を行い、開発中の計算機の性能予測や、計算機システム運用後の性能トラブルの原因究明に役立てていく。

参考文献

- [1] J.R.Larus, Efficient Program Tracing, IEEE COMPUTER, May.1993
- [2] 武内, アドレストレース駆動シミュレータの開発, 情報処理学会第51回全国大会, Sep.1995