

7C-1

スーパーコンピュータを利用した  
階層記憶シミュレーションについて

西 直樹<sup>\*</sup>、大野 直哉<sup>\*</sup>、白戸 幸正<sup>\*\*</sup>

<sup>\*</sup>日本電気(株) C&Cシステム研究所 <sup>\*\*</sup>日本電気技術情報システム開発(株)

1. はじめに

階層記憶システム(TLB、キャッシュ、仮想記憶等)の設計、評価においては、数百万以上の命令トレースあるいはアドレステレース・データを用いたシミュレーションが行われる<sup>[1]</sup>。このシミュレーションでは種々のパラメータを変えての再試行が必要とされるが、計算機の処理能力の不足、時間、コスト的な負担から十分な解析が行えないのが常である。

著者らはスーパーコンピュータを用いることにより、より大規模な問題を短時間で解析することを可能とした。本稿ではこのうちアドレス・トレース・データを入力とするシミュレーションにおけるベクトル化アルゴリズム、スーパーコンピュータSX-2上で得られた性能について報告する。

2. 階層記憶シミュレーション

階層記憶シミュレーションにおける処理の流れを以下に示す。①→(②.a or ②.b)の繰り返しに、随時③の処理が加わることになる。

- ① 入力(データ参照要求)に対して、まず一次記憶内に要求データが存在するかしないかのチェックを行う。
- ②. a 一次記憶内にデータが発見された場合(Hit時)は追い出しアルゴリズムが必要とする情報の更新を行う。
- ②. b 発見されなかった場合(Miss-Hit時)には、一次記憶から追い出しアルゴリズムに従って何れかのブロックを追い出し、Miss-Hitの原因となったブロックを取り込む。
- ③ 性能評価のためのシミュレーションであるから①,②,③の事象を観測するための処理も行う。

入力レコードとしては参照要求アドレスを直接用いてもよいが、大規模シミュレーションでは大量のトレース・データを保持する手段が問題となるので適当な圧縮を行い、ブロック・アドレスとそのブロックに対する参照回数をペアとしたものを入力レコードとしている。もちろん圧縮はシミュレーション結果に影響が出ない範囲で行う。

このようなシミュレーションをベクトル化という

観点で見ると、大きな効果を得るためには入力列方向にベクトルを取り、複数のデータ参照要求を同時に処理する必要がある。しかしオリジナルなイメージでのこのシミュレーションは一人力毎の状態遷移を基本としており、入力列ベクトルの要素間に不適当な依存関係が存在してうまくいかない。そこで、以下に示すようにアルゴリズムのレベルに戻って再検討を行った。

3. ベクトル化

ベクトル化の観点から注目すべき事実は、通常の使用環境では、一次記憶にHitしない確率は全入力の高々 $10^{-4}$ のオーダーでしかないことが上げられる。従って、Hit時の処理をいかに高速化するかが全体から見ても高速化の鍵となる。またMiss-Hit時の処理は本質的にベクトル化不能なものを含んでおり100%ベクトル化することが出来ないこともこのシミュレーションの特徴である。

開発したシミュレータでは以下の処理過程をベクトル化している。(図1にデータ構造、図2に処理の流れを示す)

①Hitするかしないかの判定

参照要求のあったアドレスをブロック化してそのまま添字として使用(リストベクトル)。従って、ページ・テーブルは二次記憶に確保されるページ数と同じだけのエントリを持っている。

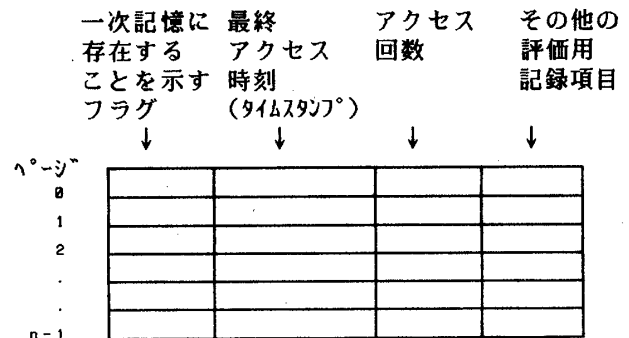


図1 ページ・テーブルの構造

Vectorization Techniques for Storage Hierarchy Simulation

Naoki NISHI<sup>1</sup>, Naoya OHNO<sup>1</sup>, Yukimasa SHIROTO<sup>2</sup>

1.NEC Corporation, 2.NEC Scientific Information System Development

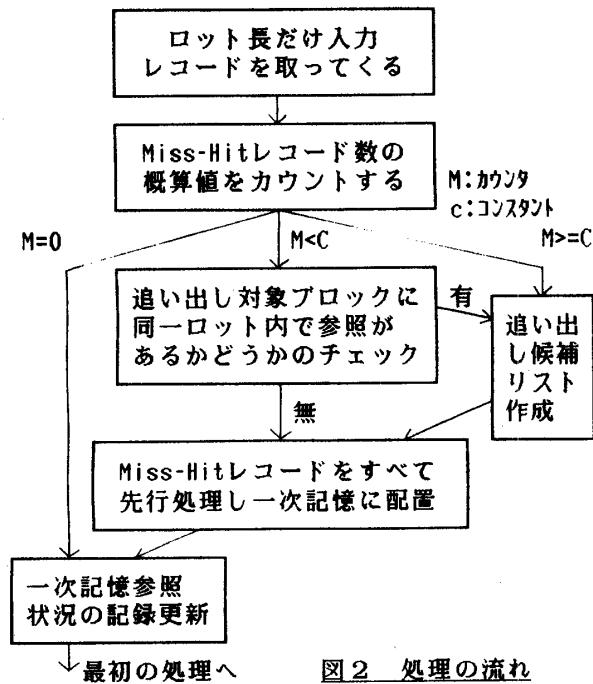


図2 処理の流れ

②参照状況の記録更新と保持

一次記憶に保持されているブロックに対して、参照の順序関係及び参照回数を記録保持する。FIFO、LFU、LRUといった追い出しアルゴリズムが追い出すブロックを決定する処理は、実際にMiss-Hitレコードを処理する時点まで遅らせる。

③Miss-Hit時の処理の一部

ベクトルとして同時に処理される入力列を以下ロット呼ぶことにする。ロットに含まれるレコード数はベクトル長に等しい。前記②のベクトル処理はロット内にMiss-Hitを引き起こす参照要求が一つも含まれない場合に可能な処理であるが、そのような確率は全入力レコードに占めるMiss-Hitレコード率(10<sup>-4</sup>オーダ)より一般に大きい。従って、ロット内にMiss-Hitレコードを含む場合の処理を高速化することも重要である。ここでは以下に示す2つの仮定が満足されるならば高速な処理を可能としている。

- ロット内に含まれるMiss-Hitレコード数が一定数よりも少ない。
- ロット内のMiss-Hitブロックを一次記憶にいれることにより追い出されるブロックに対して同一ロット内では参照要求がない。

この2条件が満たされない場合には、追い出しアルゴリズムに対応した追い出し候補リストを作成した後、実際に追い出すブロックを逐次決定する。他方、条件が満たされる場合には、直ちにMiss-Hitレコードの処理を行い、仮想的にMiss-Hitレコードがなかったものとしてベクトル処理を行う。この場合Miss-Hitレコード処理のために追い出し候補リストは作

使用計算機	一次記憶/二次記憶比率			
	2.5%	5%	10%	15%
SX-2(v)	16.77	15.61	7.70	7.02
SX-2(s)	145.33 (8.7)	143.98 (9.2)	84.39 (11.0)	73.90 (10.5)
汎用機 (6MIPS)	822.45 (49.0)	832.40 (53.3)	453.49 (58.9)	372.90 (53.1)

括弧内はSX-2(v)との処理時間比 (単位SEC)

表1 CPU時間比較結果

成せず、MAX/MIN等を用いて追い出し候補を決定する。

4. 評価

現在得られているシミュレータの能力を表1に示す。これは分子計算アプリケーションのあるフェーズをトレースしたものに対する解析時の結果である。アドレス・トレース数は127.68M。圧縮されたイメージでは1.83Mレコードになっている。表1からわかるように、SX-2上のベクトル版シミュレータとスカラ版シミュレータの性能差は約10倍となっている(ベクトル化率は86%)。また著者が従来より使用している6MIPS相当の汎用機と比較すると50倍程度の高速化を達成している。なおSX-2上のスカラ版及び汎用機ではスカラ計算機用にオブティマイズしたアルゴリズムを使用している。

5. まとめと課題

階層記憶シミュレータのベクトル化について報告した。シミュレータ自身の性能についてはスカラ対ベクトルで10倍程度であり、若干物足りない気もするが、この結果はSX-2のスカラ処理の高速性をも示していると考えている<sup>[2]</sup>。今後はさらに非ベクトル化部分の削減、一層の高速化に取り組んで行く予定である。またシミュレーションがこれほど高速になると、実用上はトレース・データの作成の方が負担であり、この面での改善を図る必要性を感じている。試作したシミュレータは、現在主に科学技術計算プログラムの特性評価に用いており、この評価結果についても報告して行きたいと考えている。(参考文献)

[1]J.Gecsei et al. "Evaluation Techniques for Storage Hierarchies," IBM Sys. J. Vol. 2, No. 2, pp.78-117, 1970.  
 [2]ラウル・メンデス,"日電ス・ハ・コンピュータSX-2の性能評価,"日電コンピュータ, No. 125, July 1986.