計算機アーキテクチャ 80－1
（1990．1．25）

密結合マルチプロセッサ記憶階層性能評価手法

長坂 充　黒川 洋　栗山 和則　和田 健一
(株)日立製作所中央研究所

大型計算機のマルチプロセッサでのオンライン･トランザクション処理では、記憶制御に関するオーバヘッドの削減が性能向上のために重要な課題となっている。私たちは、記憶階層制御のオーバヘッドを精度良く予測評価するために、トレース駆動のシミュレータを開発している。特徴は以下の通りである。(1)マルチ･トランザクション環境を実現するためにタスク･スケジューリングのシミュレーションを行っている。(2)パラメタにより、三階層記憶までの種々の構成および一致制御方式をシミュレーションできる。

ハードウェア･モニタによる実測との精度検証と、各種記憶階層構成および一致制御方式を変えて評価した結果について述べる。

Performance Evaluation Method for Memory Hierarchy of Tightly Coupled Multiprocessors

Mitsuru Nagasaka Hiroshi Kurokawa Kazunori Kuriyama Ken'ichi Wada

Central Research Laboratory , Hitachi Ltd.

1-280 , Higashi-Koigakubo , Kokubunji-shi , Tokyo 185 , Japan

When many transactions are executed on a large multiprocessor system , it is important to reduce the time for cache control. We present a simulation system using a trace-data to evaluate the time precisely in the early stage of designing the multiprocessor system. It is characterized by simulating a multi-transaction environment by means of task-scheduling , by varying the configuration of the memory hierarchy up to three and cache coherency control. We discuss the verification of the error by a hardware monitor measurement , and the evaluation varying the configuration of the memory hierarchy and cache coherency control.

1. はじめに

汎用大型計算機においては、オンライン・トランザクション処理の急増から、ますます高速の処理能力と高スループットが要求されるようになってきている。これらに対して論理方式技術、半導体・実装技術の両面から高速化の検討が進められている。

論理方式技術においてはMIEC(Mean Instruction Execution Cycle 平均命令実行サイクル数)の短縮のために様々な方式が取り入れられて高速化が図られている。近年、マルチプロセッサ化に伴い、このMIECの中で記憶階層制御に関するオーバヘッドが大きな割合を占めるようになってきており、MIECの短縮のためにオーバヘッドの削減が重要な課題となっている。しかしながら、マルチプロセッサについては評価手法が不十分であり、特にオンライン・トランザクション処理における記憶階層制御オーバヘッドを精度良く評価することは、非常に難しいものとなっている。

本稿では、実測と従来のシミュレーション結果の比較・検証を行い、シミュレータの問題点を解析した結果と、記憶階層制御オーバヘッドを精度良く評価する手法の概要と精度検証結果および各種記憶階層構成の評価結果を報告する。

2. 従来のシミュレータ

2.1 シミュレータの概要

従来のシミュレータの構成を図1.に示す。

入力としては、初期設定用およびシミュレーション用の命令トレースを用いている。

シミュレーションモデル化範囲としては命令を実行するIP(Instruction Processor)、アドレス変換バッファTLB(Translation Lookaside Buffer)、バッファ記憶BS(Buffer Storage)、ワーク記憶WS(Work Storage)からなる。IPのモデルは命令トレースから一命令ずつ参照して記憶階層へのリクエストを生成する。各バッファのモデルは、リクエストに対応するデータが登録されているかどうかを調べ、無かった場合には登録を行い、その回数を集計する。特に、TLBへの登録回数であるアドレス変換(AT)回数、BSへのデータ転送であるブロック転送(BT)回数、WSへのデータ転送であるライン転送(LT)回数、WSからMSへのデータ書き戻しであるラインバック(LB)回数を性能評価用のデータとして用いている。

パラメタとしては、転送サイズ、カラム数、ロウ数、リプレースメントアルゴリズム、ストアスルー/ストアイン等を指定できる。また、各アクセス要因ごとに参照するバッファを指定できる。

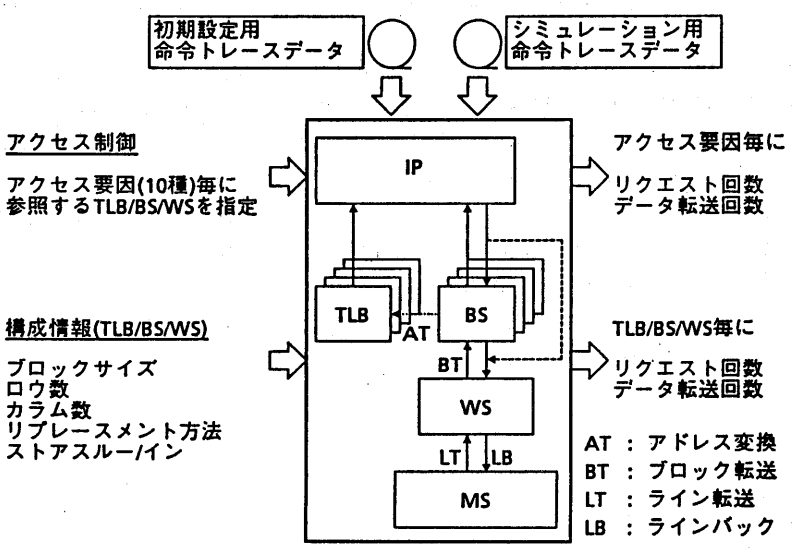

図1. 従来のシミュレータ

2.2 シミュレータの問題点

ユニプロセッサにおけるベンチマークプログラムの評価に対しては、十分使用できるシミュレータといえるが、表2.に示すように実機とシミュレータでは異なっている部分があり、これが精度低下の原因となっている。

#	項目	実機	シミュレータ
1	トランザクション	マルチ	シングル
2	BS/WS一致制御	有	シミュレーション無
3	AT	有	シミュレーション無
4	I/O	有	シミュレーション無
5	メモリのアクセス順序	パイプライン制御有	パイプライン制御無
6	BS/WSの登録/参照アドレス	実アドレス	空間番号と仮想アドレス

表2. 実機と従来のシミュレータの違い

これらのうち、BS/WSの一致制御を除いて、その他の実機との違いがシミュレータの精度に及ぼす影響をいくつかのトレースにより調べた。

トレースとしてはOS動作の一部、オンラインジョブ、ランダムなアクセスパターンを生成するように作成したシンセティックジョブを用い、M-680Hでのハードウェアモニタによる実測結果との比較を行った。比較した結果を表3.に示す。

項目	OS	Online	Synthetic(1)	Synthetic(2)
AT回数	1.02	0.78	1.00	0.47
BT回数	0.81	0.94	0.85	1.00
LT回数	1.56	0.40	1.00	1.00
I/O命令比率	0.3%	0.01%	0%	0%

ハードウェアモデル ： M680H　注) 実測との相対値

表3. 従来のシミュレータと実測との比較

まず、OS動作では、I/O命令が多いためにBT回数、LT回数とも誤差が大きくなっている。オンラインジョブでは、I/O命令は少ないが、マルチトランザクションに対応していないために、特にLT回数の誤差が大きくなっている。シンセティックジョブ(1)では、I/O命令が無くても、AT回数が多いとBT回数の誤差が大きく、シンセティックジョブ(2)では、I/O命令が無く、AT回数が小さければBT回数、LT回数とも精度よく評価できることがわかった(AT回数の誤差が大きいのは、絶対値が非常に小さいため)。

以上より、タスク･スケジューリングのシミュレーションおよび記憶階層間の一致制御のシミュレーション機能を追加して、特にオンラインジョブについて精度の検証を行うこととした。また、その他の違いについては、精度上それほど問題ないと考え、ほぼ従来のシミュレータのままとした。

3. マルチプロセッサ評価シミュレータの概要

3.1. 全体構成

図4.に新たに機能拡張したマルチプロセッサ評価用シミュレータ(MUSES)による評価方法の概要を示す。

まず、実計算機で命令トレーサにより、1トランザクション分のトレースを採取する。トレースには実行された命令の情報、実行された空間の情報、割り込み情報、仮想記憶におけるモジュールの情報等が含まれている。

これをもとに、変換および解析ツールを用いてトランザクションの中でユーザのアプリケーション空間の空間番号のみを変えたトランザクション数分のトレースと仮想空間のレイアウト情報であるモジュールマップを作成する。

以上の情報をもとにソフトウェアシミュレータがパラメタで指定されたIP台数分のトレースを生成する。そして、各IPに対応したトレースとモジュールマップを入力としてハードウェアシミュレータが記憶階層のシミュレーションを行う。

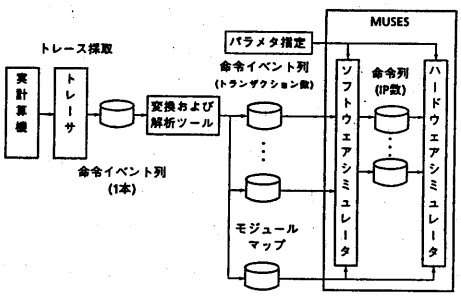

図4. MUSESの全体構成

3.2. ソフトウェアシミュレータ

ソフトウェアシミュレータはタスクのスケジューリングのシミュレーションを行い、各プロセッサで実行されるべき命令列を生成する。

図5. を用いて動作の概要を説明する。

用意されたトランザクション数t本のトレースのうち、IP台数i本分が実行中であり、残りの(t−i)本のトレースは停止中である。停止中のトレースは停止位置の状態により、TCB (Task Control Block) あるいはSRB (Service Request Block)として、停止位置の命令番号も含めてディスパッチの優先順位を管理するディスパッチング・キューモデルに登録する。

ディスパッチの順序としては、通常、グローバルのSRB、最も優先順位の高いASCB(Address Space Control Block)に対応するローカルSRB、TCBの順である。

ソフトウェアシミュレータは、実行状態のトレースでタスク・スイッチのイベントを検出すると、ディスパッチング・キューを調べて、最も優先順位の高いSRBまたはTCBに対応するトレースを実行状態にし、今まで実行していたトレースは、その時の状態に応じてSRBまたはTCBとしてディスパッチングキューに登録して停止状態にする。

以上のようにして、t本のトランザクションのトレースから各IPで実行されるべきトレースi本を生成する。

従って、トレースを採取した環境がどのようなものであっても、現状では実現されていない台数分のマルチプロセッサのトレースを評価用に生成することが可能である。

また、用意するトランザクションの本数を変えることで負荷(ワーキングセット)を変えることもできるし、タスク切替えのイベントの頻度の変更やタスクスケジューリング方法を変更することで環境の変更あるいはOSの変更により性能がどう変わるかも実際にプログラムを作成しなくても評価できる。

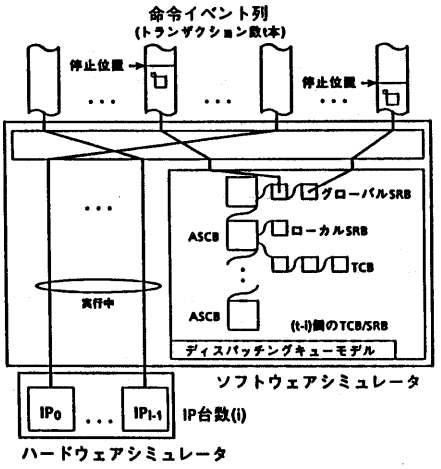

図5. ソフトウェアシミュレータ

3.3. ハードウェアシミュレータ

入力としては、ソフトウェアシミュレータの出力である各プロセッサ対応のトレースと仮想空間のレイアウト情報であるモジュールマップを用いている。

各ユニプロセッサごとに独立な部分は従来のシミュレータと同様の処理をしているが、マルチプロセッサのために、プロセッサごとに順番に1命令ずつシミュレーションを繰り返している。さらに、従来の機能に共通領域の情報によるアドレス競合の判定およびモジュールマップによりモジュールごとのより詳細な性能情報を出力可能としている。

さらに、BS/WS間の一致制御、WS/WS間の一致制御等の三階層までの記憶構成における各種記憶一致制御方式をシミュレーション可能としている。

三階層記憶における一致制御方式の例を図6. を用いて説明する。

記憶階層における一致制御は、同じアドレスに対するデータのコピーがバッファ間に存在する場合に、バッファ間でデータの不一致が起きないようにすることである。あるバッファの内容が変更された場合には、バッファ間でデータの不一致が生じる可能性があるために記憶階層間で何らかの一致制御が必要となる。

図は、記憶階層間の一致制御方式のキャンセル方式の一例を示したもので、構成はM684Hである。IP0がBSおよびWSの内容を変更した場合に、変更されたWSからBSおよび他WSに対してキャンセル要求が出て、データがキャンセルされる。これにより、バッファ間のデータの不一致は解消される。

キャンセル方式でも様々な状態を保持することにより、問合せの回数を減らす種々の方式がある。その他に変更したデータをそのまま他のバッファにも転送するブロードキャスト方式もある。

このハードウェアシミュレータでは記憶階層の構成の他にこれらの各種一致制御方式をパラメタで指定することにより、簡単にシミュレーション可能としている。

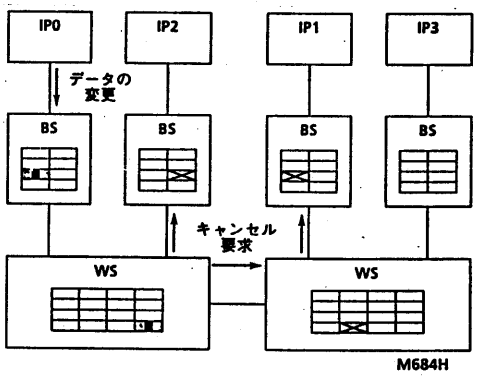

図6. 記憶一致制御

4. 評価精度の検証

図7. にハードウェアモデルとしてM68Xを用いた場合に、あるオンライン・トランザクション処理についてハードウェアモニタによる実測と比較した結果を実測との相対値で示す。

従来のシミュレータでは、WSに関するLT回数およびLB回数が実機に比べてかなり小さい値であったが、タスクスケジューリングのシミュレーションにより、マルチトランザクション環境を実現したことで評価精度をかなり改善できている。

また、マルチプロセッサに関しても、マルチ・トランザクション環境の実現と記憶一致制御のシミュレーションにより、精度よく評価できる見通しが得られた。

ハードウェアモデル ： M68X
トレース ： オンライントランザクション

項目	従来	MUSES		
	M680H	M680H	M682H	M684H
BS0系BT回数	0.88	0.92	0.89	0.90
BS1系BT回数	0.97	1.02	1.05	1.03
BT回数(Total)	0.94	0.99	1.00	1.00
LT回数	0.40	0.85	0.93	0.99
LB回数	0.12	0.96	0.99	0.99

注) 値は実測との相対値

図7. シミュレータと実測との比較

5. 各種パラメタを変化させた場合の評価結果

5.1. ブロックサイズとBT回数

IF(Instruction Fetch) とOF(Operand Fetch) を別々のBSに対して行う方式についてそれぞれのBT回数をブロックサイズを変えて測定した。ロウ数は全て4で評価している。

(1)IF用BSのブロックサイズとBT回数の関係

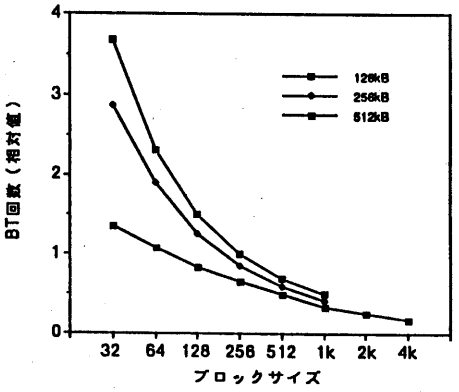

図8. IF用BSのブロックサイズとBT回数の関係

従来から、よく言われているように、IFにおけるBT回数はブロックサイズを大きくすると何れの容量においても減少する。特に、容量が小さいときには顕著である。これは、IFが連続領域をアクセスする傾向があるためである。

(2)OF用BSのブロックサイズとBT回数の関係

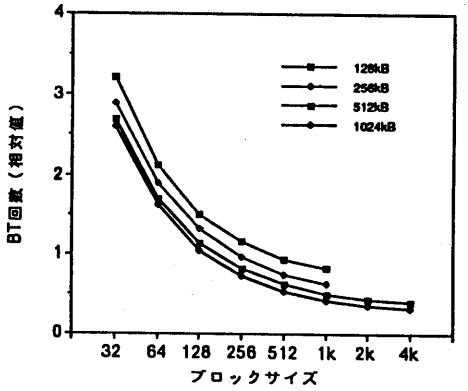

図9. OF用BSのブロックサイズとBT回数の関係

OFにおけるBT回数も大規模なオンラインジョブにおいてはIFにおけるBT回数と似た傾向を示す。特に、IFにおける容量が小さい場合に傾向が似ている。ただし、IFに比べて容量による差は小さくなっている。これは、IFに比べてOFの方がワーキングセットが大きいためであるが、オンラインジョブの傾向として従来考えられていた以上にOFにおいても連続アクセスの傾向が見られる。

(3)BSのブロックサイズとBT回数の関係

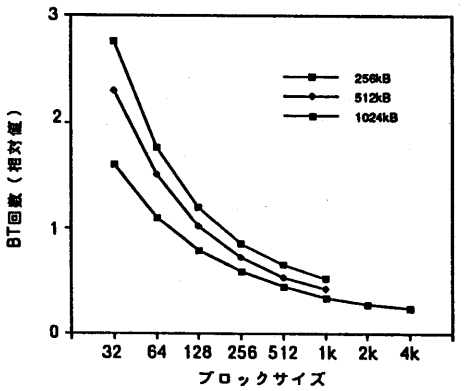

図10. BSのブロックサイズとBT回数の関係

IFとOFを合わせると、ほぼ中間的な傾向になる。これからも、アプリケーションの傾向としてワーキングセットが大きい場合には、ブロックサイズを大きくすることでトータルとしてBT回数をかなり減少させることができることがわかる。

5.2. ブロックサイズとオーバヘッドの関係

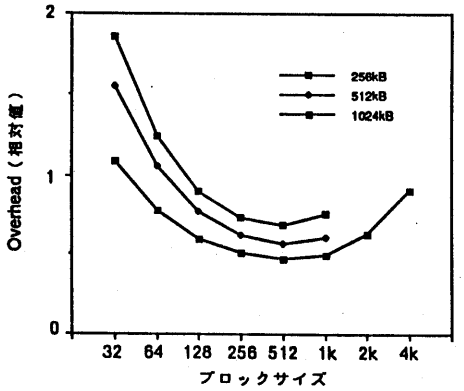

図11. ブロックサイズとオーバヘッドの関係

ブロックサイズが大きくなると、転送時間も増加する。ただし、転送時間は起動時間と転送量に比例する部分からなり、ブロックサイズが2倍になったからといって転送時間も2倍になるわけではない。

現在の大型計算機における転送時間を仮定してMIECに占めるオーバヘッドを求めた結果を図11に示す。このオンライントランザクションプログラムでは、ブロックサイズが256B~1kB程度の場合に極小になっている。すなわち、プログラムの大規模化に伴い、従来実現されているブロックサイズよりも大きいブロックサイズにおいて性能の最適点があることがわかった。

5.3. 容量とBT回数の関係

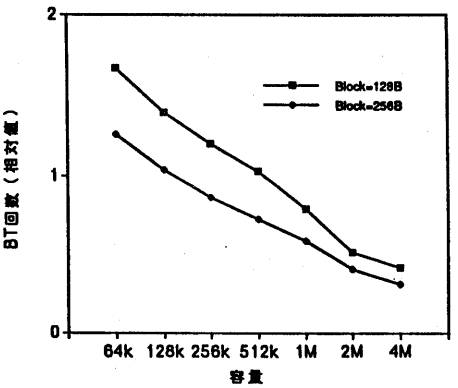

図12. 容量とBT回数の関係

容量を増やすほどBT回数は減るが、ブロックサイズを大きくした場合のBT回数の減少分と比較すると、容量の効果の方が小さいことがわかる。従って、転送時間とコストを考えて、容量とブロックサイズによる最適点を見つける必要がある。

5.4. CPU構成とBT回数

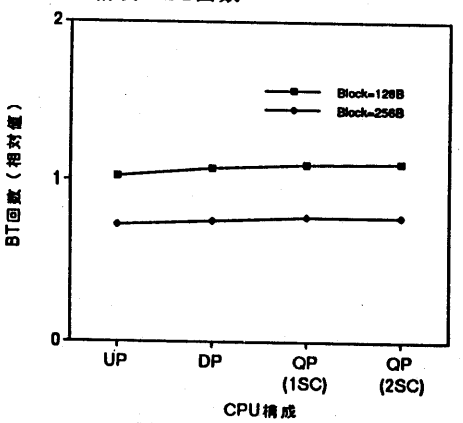

図13. CPU構成とBT回数

CPUの構成をUP、DP、WS(SC)が1つの場合のQP、WS(SC)が2つの場合のQPについてシミュレーションした。BT回数は幾分増加するのみである。これは、ワーキングセットに比べてBSの容量が小さいために、BS間のキャンセルが起きてBT回数が増加する確率が小さいことを示している。

5.5. ラインサイズとLT回数の関係

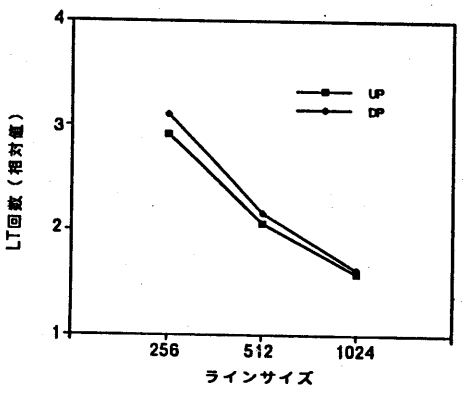

図14. ラインサイズとLT回数の関係

ラインサイズもブロックサイズと同様に256B~1024Bの範囲ではラインサイズを大きくするほどLT回数は減少する。ただし、性能を考える場合には転送時間まで考慮しなければならない。

5.6. 容量とLT回数の関係

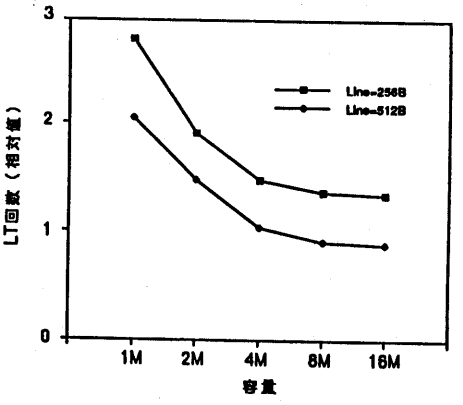

図15. 容量とLT回数の関係

ラインサイズが256Bの場合でも512Bの場合でも容量を増やすほどLT回数は減少する。ただし、容量が4MBを越えるとほとんどLT回数は減少しなくなる。

5.7. ロウ数とLT回数の関係

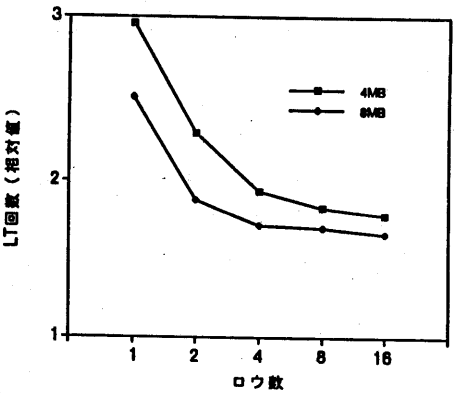

図16. ロウ数とLT回数の関係

ロウ数を増やすと容量が4MBの場合でも8MBの場合でもLT回数は減少する。ただし、ロウ数を8以上に増やしてもほとんどLT回数は減少しない。

5.8. CPU構成とLT回数

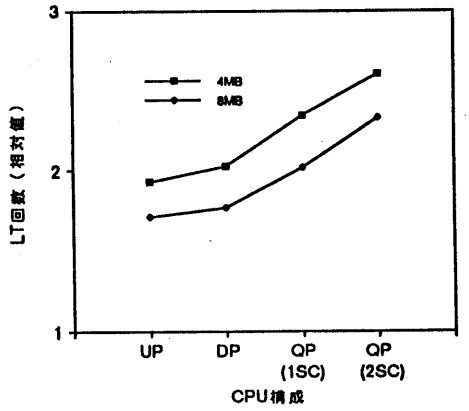

図17. CPU構成とLT回数

CPUの台数が増えるとBTはあまり増加しなかったが、LTはかなり増加する。特に、WS間の一致制御のためにWSが2つの場合にはWSが1つの場合に比べてかなりLT回数が増加する。

6. おわりに

記憶階層制御のオーバヘッドを精度良く予測評価するために、トレース駆動のシミュレータを開発した。その特徴は以下のとおりである。

(1)マルチ・トランザクション環境を実現するためにタスク・スケジューリングのシミュレーションを行っている。

(2)タスクのスケジューリング方法を変更してシミュレーションできる。

(3)パラメタにより、三階層記憶までの種々の構成および一致制御方式をシミュレーションできる。

(4)ソフトウェアのモジュールごとの性能解析が可能である。

ハードウェア・モニタでの実測との精度検証を行った結果、十分各種方式を評価できる精度が得られた。

また、各種パラメタを変更して評価した結果、大規模オンライン・ジョブでは、ブロックサイズの最適点が256B~1024Bにあること、ロウ数を8以上に増やしても性能向上には余り寄与しないこと、容量よりも転送サイズを増やした方が性能向上の可能性があるという結論が得られた。

今後は、さらに評価精度向上のために、パイプライン・シミュレータとの連動による評価を検討している。

参考文献

[1]松岡　浩司, 堀川　隆, 難波　信治:マルチプロセッサシステムの評価技法と評価システム, 情報処理学会, OS研究会, Feb.(1989)

[2]R. T. Short , H. M. Levy : A Simulation Study of Two-Level Caches ,Proc. 15th Annual International Symposium on Computer Architecture(1988), pp. 81-88

[3]J. L. Baer , W. H. Wang : On the Inclusion Properties for Multi-Level Cache Hierachies , Proc. 15th Annual International Symposium on Computer Architecture(1988), pp. 73-80

[4]A. Agarwal , R. L. Sites , M. Horowitz : ATUM:A New Technique for Capturing Address Traces Using Microcode, Proc. 13th Annual International Symposium on Computer Architecture(1986), pp. 119-127

[5]R. L. Sites , A. Agarwal: Multiprocessor Cache Analysis Using ATUM, Proc. 15th Annual International Symposium on Computer Architecture(1988), pp. 186-195