3J−6　バーチャルタイムによる並列論理シミュレーション

松本 幸則　　瀧 和男
(財) 新世代コンピュータ技術開発機構

1　はじめに

論理シミュレーションはLSI設計工程の中で設計仕様の検証に重要な役割を果たしているが、膨大な時間が費やされてしまう点が大きな問題となっていることから、高速シミュレータに対する要求は強い。ハードウェアエンジンを用いた高速シミュレータはこの要求には答えるが、柔軟性の面で問題があり、高精度のシミュレーションにはソフトウェアによる並列論理シミュレーションが有望な方法として考えられる。

我々は第五世代コンピュータプロジェクトの一環として、並列推論マシン実験機「Multi-PSI」上にバーチャルタイムによる並列論理シミュレーション実験システムを構築した。本論文では、まず、システムの概要を述べ、プロセッサ毎のローカルなメッセージスケジューラおよびアンチメッセージの削減について説明する。続いて処理性能に大きく影響する負荷分散戦略について論じる。ここでは実用的な回路分割の戦略として、縦割り指向戦略と名付けた回路分割戦略を提案する。最後に、この回路分割戦略を用いた場合の実際の性能、スピードアップについての計測結果を報告し、本戦略の有効性とシステムの絶対性能の高さを示す。

2　事象シミュレーションとバーチャルタイム

本システムのシミュレーション方式である事象シミュレーションは、複数のオブジェクトがメッセージ通信を行なうことによって、次々と状態を変えていく形にモデル化できる。オブジェクトは状態オートマトンとして表現され、メッセージは事象情報を持つとともに、事象の発生時刻がスタンプされている(タイムスタンプ)。

このモデルでは、逐次実行の場合タイムホイルと呼ばれるメッセージの時刻集中管理機構をおくのが一般的である。しかし、分散メモリ環境での並列事象シミュレーションでは、集中管理機構は並列性を損なう大きな要因になることから、ローカルな時刻管理機構が望ましい。

このような方法として、Chandy, Misraらは各オブジェクトでのメッセージ待ち合わせ機構に基づく方法を提案している[2]。この方法は効率的なものであるがデッドロックの危険性を持ち、その回避のためのオーバヘッドが問題になっている。

これに対し、Jeffersonは、バーチャルタイムの概念に基づくタイムワープ機構(以後バーチャルタイム方式と呼ぶ)を提案している[1]。バーチャルタイム方式では、各オブジェクトは、メッセージは正しい順序に到着するという仮定の基にメッセージ及び状態の履歴を保存しながら処理を進め、到着順序の矛盾を発見したところで履歴を巻き戻し、処理のやりなおしをする(ロールバック)。さらに、その時点で誤って送信したことが判明したメッセージに対しては、メッセージを取り消す意味を持つアンチメッセージなるものを送る。

ロールバックの多発は処理を遅らせる原因と考えられるため、その発生を抑えるための工夫が重要となってくる。

3　実験システム概要

本実験システムは、並列論理型言語KL1で記述され、Multi-PSI上に実装されたシステムである。Multi-PSIはMIMD型マシンで、要素プロセッサ64台が2次元メッシュ状のネットワークで結合されている。メモリは全て分散管理されており、他のプロセッサへのデータアクセスコストは高いが、台数拡張性に富む。

本シミュレータでは、ゲートレベルの順序回路を扱う。信号値はHi、Lo、X(不定)の3値モデルとし、遅延は各ゲートに単位時間の整数倍を割り当てるような割り当て遅延モデルとする。本システムの目的は並列処理実験であることから、最低限の一般性を持たせた単純な仕様にしているが、更に機能を拡張したシステムへの移行は容易である。

3.1　ローカルメッセージスケジューラ

本システムでは、ゲートをオブジェクトに対応させ、信号線をチャネルに対応させている。この場合オブジェクトはプロセッサ数に比べはるかに多いため、各プロセッサ内でのローカルなメッセージのスケジューリングを行うことは、ロールバック発生の低減に有効である。このため、本システムでは各プロセッサ内にローカルメッセージスケジューラを置いている。

メッセージは、送信側のオブジェクトから、受信側のオブジェクトが管理されているメッセージスケジューラにまず送られ、そこでメッセージキューに登録される。スケジューラは登録中のメッセージのもつ最小のタイムスタンプ値をクロックとし、クロックに対応したメッセージを順次受信側オブジェクトに送る。

スケジューラは、基本的にタイムホイルに非常に類似した働きをする。しかし、タイムホイルでは、必ず時刻の増大方向にのみ進むのに対し、このスケジューラは時刻の減少方向に進む場合がある点が異なる。他のプロセッサからクロックより若い時刻のタイムスタンプ値を持ったメッセージが届いた場合にこの様な動作をする。

3.2　アンチメッセージの削減

本システムのシミュレーション対象である論理回路は静的ネットワークであり、一つのチャネル上のメッセージの送信者と受信者は固定されている。また、KL1では、同一ストリーム上のメッセージの送信順が保存されているため、送信側オブジェクトとメッセージスケジューラ間、及びメッセージスケジューラと受信側オブジェクト間ではメッセージの送信順序は保存されている。しかし、スケジューラ内でタイムスタンプ値によるソートを行っているため、このままでは送信オブジェクトと受信オブジェクト間でのメッセージ順序の保存ができない。このため、チャネル毎に各メッセージに対してオーダースタンプを行うことでメッセージの送信順の保存を行っている。このシステムでは、以下のようにアンチメッセージを削減することができる[3]。

送信側オブジェクトでロールバックが発生した場合、本来は無効とすべきメッセージ全てに対しアンチメッセージを送らなければならないが、本システムでは無効とすべきメッセージのうち、最小のタイムスタンプ値を持つメッセージに対応したアンチメッセージのみを送る。この時、同一チャネル上でアンチメッセージ以上のタイムスタンプを持ち、且つ、アンチメッセージより小さいオーダースタンプを持っているメッセージは全て無効とすべきメッセージであり、また、これ以外に無効とすべきメッセージはない。したがって、受信側オブジェクトは、同一チャネル上で受信したメッセージのうち、アンチメッセージ以上のタイムスタンプ値を持ち、且つアンチメッセージより小さいオーダースタンプ値を持つメッセージに対してのみ無効化の手続きを取ればよい。

4　回路分割戦略

並列論理シミュレーションは一つのメッセージ当たりの処理量が小さい、即ち小粒度であるためプロセッサ間通信量が問題になる。また、バーチャルタイム方式では、見込み計算を行っており、ロールバックの発生量も問題になる。したがって、この問題を分散メモリ型並列マシン上で実行する場合、1: 負荷の均等分散、2: プロセッサ間通信の低減、3: ロールバック発生回数の低減、の3点を目標に回路分割を行なう必要がある。この場合、上記3点を満足するような評価関数を定義し、その値を最良にする分割を求めなければならないが、このような問題は一般にNPハー

Parallel Logic-level Simulation based on the Virtual Time
Yukinori MATSUMOTO, Kazuo TAKI
Institute for New Generation Computer Technology

ドであるため、何らかのヒューリスティックスを用いる必要がある。今回、上記目標の3点をある程度満足し、かつ計算時間が実際のシミュレーション時間に比べ十分に小さいような戦略として、縦割り指向戦略と名付けた戦略を試みた。

縦割り指向戦略は、基本的に縦方向につながったゲートを探索してクラスタを切り出すことを意図している。最初に外部入力端子につながっているゲートを探索開始点候補としてキューに入れる。まず、キューから探索開始点を取りだし、その出力先ゲートで、まだどのクラスタにも属していないゲート一つをそのクラスタに取り込む。残りのゲートは別の探索開始点候補としてキューに入れる。そしてクラスタに取り込んだばかりのゲートの出力先について同様の処理を続ける。取り込めるゲートがなくなった時点でそのクラスタ生成は終了し、キューから、クラスタに未所属の新たな探索開始点を取りだし、再び以上の処理を行なう。

このようにして生成されたクラスタのうち、大きさの小さいものについてはそれがつながっている別のクラスタにマージする。また、極端に長いクラスタは幾つかに切り分ける。そして最後に、生成されたクラスタをランダムに各プロセッサに割り当てる。

縦割り指向戦略は、比較的良い分割を与え、かつ計算時間の短い戦略であることから実用的な分割戦略であると考えられる。

5 実験結果および考察

ISCAS'89で公開された順序回路s13208.benchについて、縦割り指向戦略により回路分割を行なった後、「Multi-PSI」上で論理シミュレーションを行ない、スピードアップを計測した。今回の実験ではクロック線以外の入力端子には、クロックの立ち上がりに同期してランダムに信号値が変化するような入力信号列を与えた。

表1には縦割り指向戦略分割によるプロセッサ間渡りの信号線の、全体の信号線数に対する割合を示している。表中のPEは要素プロセッサを意味している。また、表2および図1にシミュレーション実行結果を示す。表中でのロールバックメッセージ比率は、最終的にロールバックされたメッセージの、真のメッセージ数に対する割合を表す。メッセージバランスは、最も多く真のメッセージを処理したプロセッサでの処理数の、平均的な真のメッセージ処理数に対する割合を表す。真のメッセージに対する処理量が等しいと仮定すれば、この値が負荷の不均等の度合いを表す。スピードアップ上限は、負荷の不均等を考慮した場合のスピードアップの上限値: (使用プロセッサ数)/(メッセージバランス)を表す。また、図1の横軸は使用プロセッサ数を表し、縦軸はスピードアップを表す。

表1からは、縦割り指向戦略によってプロセッサ渡りの信号線数が少なくなるような回路分割ができていることが読みとれる。縦割り指向戦略では、少なくともゲート数分の信号線がプロセッサ内の信号線となると考えて良い。13207.benchの回路内信号線数は18489、ゲート数は11965であるため、プロセッサ渡りの信号線の比率は最大でも35.29%である。実際には、64台への分割の場合で17.43%と最大値に比べ小さい。この理由は、回路分割時の、一定の大きさ以下のクラスタをそれが接続している別のクラスタにマージするという手続きの効果と思われる。

図1では、プロセッサ32台まではほぼリニアで良好なスピードアップを得ることができているが、64台による実行ではスピードアップがやや悪くなっている。一般には、スピードアップの悪化理由は、プロセッサ間通信のコスト高、ロールバックの頻発、負荷の不均等、の3点が考えられる。しかしながら、表2で、プロセッサ数16台以上の場合の実際のスピードアップが、負荷の不均等を考慮した場合のスピードアップ上限にほぼ等しいことから、これらの場合は負荷の不均等がスピードアップに最も影響していると考えられる。極端に多くの負荷が割り当てられてしまったプロセッサではシミュレーション進行が常に一番遅れていると考えられるため、そのプロセッサではロールバックはほとんど発生しないが、このプロセッサが全体のスピードを決定することになる。今回の対象回路の場合、64台のプロセッサの実行時には、回路分割で生成するクラスタの大きさを更に小さくすることでランダム性を増した方が、負荷の不均等がある程度解消され、更に良いスピードアップが得られることが期待できる。なお、表中で実際のスピードアップが負荷の不均等を考慮した場合のスピードアップ上限を越えている場合がある。これは、真のメッセージに対する処理量が均一であるという仮定のもとにスピードアップ上限を求めたが、実際は多少のばらつきがあるためと思われる。

ロールバックがスピードアップに及ぼす影響について考察してみると、16台以上のプロセッサを用いた実行でのロールバックは、全体のスピードアップには影響を及ぼしておらず、常にシミュレーション進行が速すぎるプロセッサのみで発生していることが予想される。また、8台以下のプロセッサによる実行では、メッセージ通信量とロールバック双方がスピードアップに影響していると考えられるが、その影響は極僅かである。Chandyらの方法では、デッドロック回避のためのヌルメッセージ発生量が大きな問題である[4]。これに対し、バーチャルタイム方式ではロールバックの発生が問題と考えられているが、今回の実験ではロールバックの発生は少なく、スピードアップに及ぼす影響も小さい。このことからバーチャルタイム方式は並列論理シミュレーションのローカルな時刻管理機構として有望と考えられる。

一方、システムの絶対性能は約47kイベント/秒を達成しており、割り当て遅延モデルに基づくソフトウェアシミュレータとしては高性能を実現できたと考えられる。次機種の並列推論マシンPIM上では更に20倍程度の高速化を期待している。

PE数	2	4	8	16	32	64
PE渡り信号線比率(%)	10.02	14.02	15.99	16.90	17.22	17.43

表1：縦割り指向分割戦略によるプロセッサ渡り信号線比率

PE数	性能(イベント/秒)	スピードアップ	ロールバックメッセージ率(%)	メッセージバランス(%)	スピードアップ上限
1	986	1.0	—	—	—
2	1,852	1.878	1.33	102.33	1.95
4	3,545	3.595	4.05	103.27	3.87
8	7,090	7.191	7.01	106.10	7.54
16	14,331	14.534	7.89	116.37	13.75
32	28,209	28.619	11.33	118.83	26.93
64	46,666	47.329	24.65	139.09	46.01

表2：縦割り指向分割戦略による実行結果

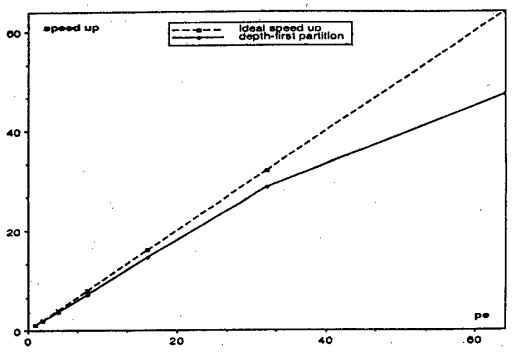

図1: スピードアップ

6 おわりに

バーチャルタイム方式による論理シミュレーション実験システムを構築し、縦割り指向戦略により回路を分割した後、性能評価を行なった。その結果、縦割り指向戦略による回路分割は、実用的な分割を得る方法であることが確認できた。また、ロールバックはシステムの性能を大きく低下させることはなく、全体としてはリニアに近いスピードアップを得ることができた。

今後は、更に別の分割戦略との比較を行なうと共に、ロールバック頻度と分割との関係について解析を進めていく予定である。

参考文献

[1] D.R.Jefferson, "Virtual Time", ACM Transaction on Programming Languages and Systems,Vol.7,N0.3, 404-425 (1985)

[2] J.Misra, "Distributed Discrete-Event Simulation", Computing Surveys,Vol.18,No.1, 39-64 (1986)

[3] 福井 眞吾 "バーチャルタイムアルゴリズムの改良", 情報処理学会論文誌, Vol.30,No.12, 1547-1554(1989)

[4] 下郡慎太郎, 鹿毛哲郎 "メッセージドリブンによる並列論理シミュレーション", 電子情報通信学会研究会報告, CAS88-110, 23-30(1989)