

並列事象型シミュレーションにおけるより楽観的な BTB 方式*

5 F - 2

鏡味 秀行

静岡大学 工学研究科[†]渡辺 尚[‡]静岡大学 情報学部[§]

1はじめに

シミュレーションは、システムを分析、検討する問題解決手法として有効である。近年、計算機技術の発展によりシミュレーションの対象とされる実世界のシステムは大規模、複雑化し、それに応えるより高速なシミュレーションが求められている。特に計算機のダウンサイジング化が唱えられると共に、従来単一のプロセッサで行っていたシミュレーションを、分割して複数のプロセッサに割り当てるこにより高速化を図ろうという、離散事象並列シミュレーション(Parallel Discrete Event Simulation 以下 PDES)が注目されるようになった。

しかし、並列シミュレーションは一般的に確率的不確定要素をふくむジョブであるために、事前に処理計画を立てることが困難で、そのためプロセッサの制御方法に十分な検討が必要である。同じ並列処理の分野でも配列データのベクトル化計算等、単一の処理を異なるデータを相手に行う制御とは明らかに対極をなす。

本研究では、並列シミュレーションの処理能力向上を目的とし、プロセッサの制御方式として楽観的手法と保守的手法の中間に位置する Breathing Time Buckets 法(BTB 法)を取り入れた並列待ち行列網シミュレータを超並列計算機上に実装し、その処理能力に関して評価し、問題性を検討する。さらに、その問題点を改善する方法の一つとして、改良版 Breathing Time Bucket アルゴリズムを実装しその評価を行う。

2 PDES の問題点

PDES は、論理時刻の更新間隔が一定ではないため、プロセッサごとに分割して実行する際にプロセッサ同士の時刻のずれが顕著に現れる。そのためシミュレーションの制御にはさまざまな問題が発生する。

例としてタイムスタンプ 10 の論理プロセス LP_1 のイベント E_1 と、タイムスタンプ 20 の論理プロセス LP_2 のイベント E_2 を考える(図 1)。2つのイベントが互いに独立しているのであれば E_1 と E_2 は同時実行可能である。しかし、もし E_1 を処理することによって、タイムスタンプが 20 より小さい新たなイベント E_3 を LP_2 において生成するのならば、 E_3 は E_2 に影響を与えるということから、 E_1 が E_2 よりも先に実行される必要がある。もし E_2 が先に処理されると、いずれは生成される E_3 は、過去に影響を与えるという起こってはならない状態を作ってしまう。同時実行が可能かどうかを事前に決定できれば良いが、それには E_1 の処理内容を調べる必要があり、一般にそれをシミュレーションを実行せずに行なうことには困難である。

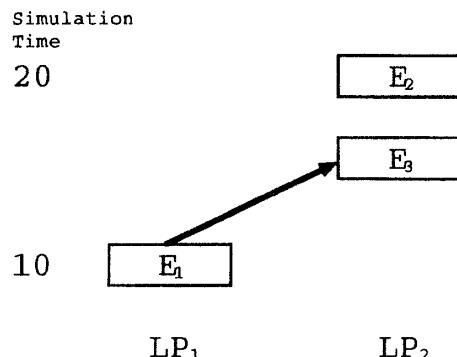


図 1: 従属関係のあるプロセス

3 PDES のアルゴリズム

PDES の問題に対しては過去に解決方法としてさまざまな手法が検討されており [1]、おもに時間的な矛盾を起こさないように同期を取りつつメッセージの送信を行う conservative(保守的)なアプローチと、一時的に時刻の矛盾の発生を許す optimistic(楽観的)なアプローチに大別される。

optimistic アプローチでは各プロセッサはあらかじめ矛盾の発生に備えて修復過程を用意しておく。各プロセッサは独立に処理を進めて行き、矛盾が起きた際に時刻をロールバックされることにより整合性を保たせる。代表的なものには、Jefferson の Time Warp、佐藤、渡辺らの先行制御方式などがある。optimistic アプローチはプロセッサ間で同期を取ることがないため、普段はシミュレーション内で並列性を最大限に生かせるということが利点である。その反面、処理履歴の保存や、時刻矛盾の発見、修復に伴う附加的な作業が必要である。

また、optimistic と conservative を複合させたアプローチもあり、Steinman による Breathing Time Bucket[2]、Breathing Time Warp [3]などが研究されている。これらの方では、イベント処理の基本動作は optimistic である一方、ロールバックの連鎖を引き起こすアンチメッセージを生じさせないように同期を取りメッセージを送信する。本研究では、Breathing Time Buckets をより optimistic な方向に改良しその性能について評価を行う。

4 改良型 BTB アルゴリズム

本研究で提案するアルゴリズムは基本的には Breathing Time Bucket アルゴリズムを踏襲したものであるが、より並列性を意識した方式である。

BTB アルゴリズムではアンチメッセージの存在を無くすため、1 サイクルの中でイベントによって生成された新しいイベントのうち、パリア同期を用いて、最も小さいタイムスタンプをもつものを GEH(Global Event

*Optimistic BTB Approach in Parallel Discrete Event Simulation

[†]Hideyuki Kagami, Takashi Watanabe

[‡]Graduate School of Engineering, Shizuoka University

[§]Faculty of Information, Shizuoka University

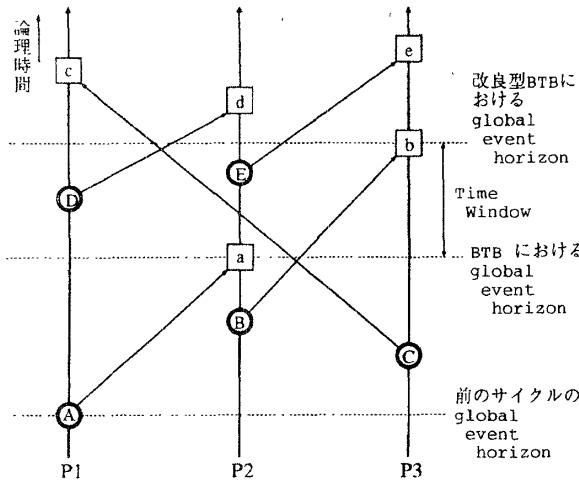


図 2: Time Window を設置した BTB のアルゴリズム

Horizon) とし、前のサイクルの GEH との間にあるイベントのみメッセージを放出することを許す。このため、GEH の時間的差分が小さい場合、バリア同期の回数の増加によるオーバヘッドが大きな問題点となる。

これに対して本研究では、GEH を起点とした time window を設置し、その中でメッセージ群を交換する改良型 BTB 方式を提案する(図 2)。この方式では time window の範囲内で伝送メッセージを有するイベントが存在した場合はロールバックによるアンチメッセージの発生が生じる(図 2 では P2 に a が到着することによって、既に処理を終えている E に時間的矛盾が起き、E は e を無効にするアンチメッセージを伝送する必要がある)。しかし、前のサイクルの GEH と新たな GEH との時間間隔を多く取ることによって一回の同期につき同時に送出できるメッセージ数を増やすことができ、より optimistic な処理を行うことができる。図 2 では従来の BTB アルゴリズムにくらべ、新たにイベント D が伝送可能となってい

5 性能評価

本研究では、改良型 BTB 方式の性能特性を検討する為に、ノード分散型の並列待ち行列シミュレータを富士通の超並列計算機 AP1000+ 上に実装した。対象となるモデルを以下に示す。

1. ネットワークはプロセッサ数と同数の 2 ~ 64 の完全網のノードからなる。
2. 各ノードへのパケットの到着率は平均 λ (個/sec) のボアソン分布、パケット長は平均 μ (sec) の指数分布に従う。

図 3 はシングルプロセッサのシミュレータに同等の処理をさせたものを比較対象とし、BTB および改良型 BTB の処理能力特性を求めた結果である。ここで、

$$\text{処理能力} = \frac{\text{シングルプロセッサでの実処理時間}}{\text{マルチプロセッサでの実処理時間}}$$

である。AP1000+への実装に関して、実際にはイベントの処理に対してバリア同期のオーバヘッドが膨大であつ

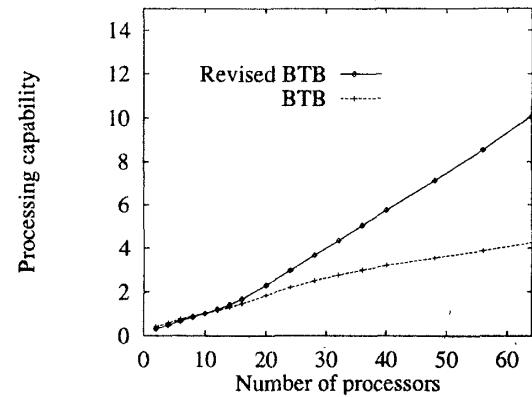


図 3: BTB および改良型 BTB の処理能力特性

た為、バリア同期のオーバヘッドが $1/10$ であると仮定したモデルを示す。また、time window の幅は 20(sec)とした。

図 3 よりプロセッサ数 10 を過ぎた辺りから time window による性能向上の効果が見られる。プロセッサ台数が少ない場合に性能向上が見られない理由は、プロセッサ同士のメッセージによるつながりが密となるが故に time window の増加によるアンチメッセージの発生する状況ができやすくなる為である。このため最適な time window の値はプロセッサ台数によって変わり、その値は、アンチメッセージの発生頻度とバリア同期のオーバヘッドのトレードオフであることが分かる。

6 おわりに

今後の課題としては、プロセッサ台数、アンチメッセージの発生頻度、及びバリア同期のオーバヘッドなどから最適な time window を求める解析法、また、バリア同期を用いずにアンチメッセージを削減する optimistic アルゴリズムを検討中である。

参考文献

- [1] R. M. Fujimoto, "Parallel discrete event simulation," *Communication of the ACM*, 33, 10, pp. 30-53, Oct. 1990.
- [2] Jeff S. Steinman, "SPEEDES: A Multiple-Synchronization Environment for Parallel Discrete-Event Simulation," *International Journal in Computer Simulation*, pp. 251-286, Feb. 1992.
- [3] Jeff S. Steinman, "Breathing Time Warp," *7th Workshop on Parallel and Distributed Simulation (PADS '93)*, pp. 109-118, May 1993
- [4] 鏡味, 渡辺, "Implementation of Parallel Discrete Event Simulation Using BTB Approach on AP1000" *Parallel Computing Workshop (PCW'96 Japan)*, pp. P1-G