

シミュレーション先読みによる PDES の実行効率向上

永田 和敏、成田 誠之助
早稲田大学理工学研究科

PDES の実行効率に影響を与える要素として、仮想時刻同期手法とマッピング手法が挙げられる。本論文では乱数をコントロールすることによって、シミュレーションを先読みし PDES の実行効率を向上させる。仮想時刻同期手法においては同期法の同期時間をシミュレーション先読みによりメッセージポイントを探り、最大限まで長くする事によって同期コストを削減する。また、動的マッピングにおいて、我々は GA を用いてきたが、GA における計算時間の長さによる効率の低下を考え、シミュレーション先読みにより得られた統計データを用いることにより、長期的な GA、短期的なマッピング、両方面から動的マッピングを実現することに成功した。

Application of reading simulation ahead to Parallel Discrete Event Simulation

Kazutosi Nagata, Seinosuke Narita
Department of science and engineering, Waseda University

There are two major elements which influence the execution efficiency of PDES. One is a synchronization method, and the other is a mapping method.

This thesis proposes dynamic synchronization method using reading simulation ahead. We use this method, and a synchronous interval can taken long. As a result, this synchronous cost can be reduced. In addition, we had this method adjust to a dynamic mapping. An efficient simulation was able to be achieved as a result.

研究背景

年々、ネットワークシステムを初めとする様々なシステムが大規模化、複雑化の道を進んできている。それにより、シミュレーションの分野においても大規模なモデル、複雑なモデルを扱うものが増え、その計算時間の短縮というのが大きなテーマの一つとして挙げられるようになった。シミュレーションを高速に行うため

の方法として、コンピュータの処理速度を向上させることが考えられるが、これは費用および時間に多大なコストを要し、また、その時代における限界というのがある。そこで、複数のコンピュータを用いて並列処理を行う並列シミュレーションが注目を浴びるようになった。その中でも特に離散事象シミュレーションを扱ったものを並列離散事象シミュレーション (PDES)

といい、我々、成田研究室シミュレーション班は、この PDES の高速化を目的として研究を行ってきた。

PDES

並列離散事象シミュレーションにおいて、その実行効率に影響を与える要素として仮想時刻同期手法およびマッピング手法が挙げられる。仮想時刻手法はシミュレーションの正当性を保ちながらデッドロックに陥ることなく効果的にメッセージを送るための手法である。また、マッピング手法は各ノードをプロセッサに割当てるための手法であり、事象処理数の均等化、および、メッセージ数の削減を目的とする。

仮想時刻同期手法として、従来の同期的手法において、保証時刻を決定する際、乱数をコントロールする事により先に起こるシミュレーションを先読みし、メッセージが発生する時刻を取得する事で、保証時刻を最大限に長く取る手法を採用した。これにより同期コストの削減、通信コストの削減、各 PE のアイドル時間の削減を実現し、並列アルゴリズムのボトルネックの部分削減する事でシミュレーションの高速化を目指す。

また、マッピング手法においては遺伝的アルゴリズム (GA) を用いた動的マッピング手法を適応するとともに、先読みシミュレーションで得られた統計データをもとにマップを更新する事で、GA の計算時間の長さによる効率悪化に対して対応させた。これにより、長期的な動的マッピングには GA を、短期的な動的マッピングには先読みシミュレーションで得られた統計データでの動的マッピングを採用する事によりシミュレーションの効率的なマップを作成する。

仮想時刻同期手法

PDES 仮想時刻同期手法は各 PE の仮想時刻進行の歩調を取るものである。これにより独立して計算を行う各 PE の事象処理中に発生するメッセージ事象の処理の制御を行う。このメッセ

ージのタイミングを間違えると、事象処理中に仮想時刻の逆転が起こってしまうシミュレーションの信頼性を損なう結果となってしまう。現在までに制御手法として保守的手法では、Null Message 手法、問い合わせメッセージ手法、同期法等が提案され、楽観的手法では TimeWarp 法などが提案されてきた。本研究では保守的手法のなかで、効率的で、比較的高速なシミュレーション結果を算出してきた同期法の弱点を補う事で、更なるシミュレーション効率の向上を目指した。

同期法

同期的手法は、バリア同期を行いながらシミュレーションを進めていく並列アルゴリズムである。同期法のバリア同期までの間隔は固定となり次のようなアルゴリズムとなる。

- ・まず、各 PE が他の PE に対して依存関係を持たないですむ時間間隔 T を求める。これを最小保証時間という。
 - ・次に現在の仮想時刻を T_c (先頭事象時刻) とすると、全 PE における各 PE が独立性を持って事象を処理できる時刻 T_s を算出する。
$$T_s = T_c + T$$
 - ・次に各 PE で算出した上記 T_s の最小値を計算する。
$$T_{s_min} = \min(T_{s1}, \dots, T_{sn})$$
- 以上この処理の繰り返しでシミュレーションは進んでいく。

シミュレーション先読み同期法

以前の同期法には無駄が存在する。これを図 1 を用いて説明する。図 1 の 2 回目のバリア同期をみるとバリア同期時に何の処理も行われていない。これはその前の事象処理の時点でメッセージに相当する事象が発生していないためである。つまり無駄な同期を行っていることになる。この同期の無駄がシミュレーション中にどのくらい存在するのかの統計を取ってみた。図 2 を見ても無駄の大きさがわかる。並列シミュレーションにおいては、いつメッセージが起こ

るか分からない。そこで、同期法では保守的な時間間隔でバリア同期を行うことで、これを解消しているわけであるが、ここに欠点があった。もし未来のシミュレーションを垣間見る事が出来、メッセージが発生するポイントが分かれば、無駄なバリア同期を削減する事が出来るものと思われる。これまで、我々の研究から並列シミュレーションにおいて PE のバリア同期コストがメッセージ送信コストに比べても無視できないほど大きい事が分かっている。よって、このシミュレーション先読み同期法を採用し、今までの固定された時間間隔ではなく、自らシミュレーションを先読みし動的に時間間隔を入手していくアクティブなアルゴリズムを採用する事で、大きな効果があげられると思われる。

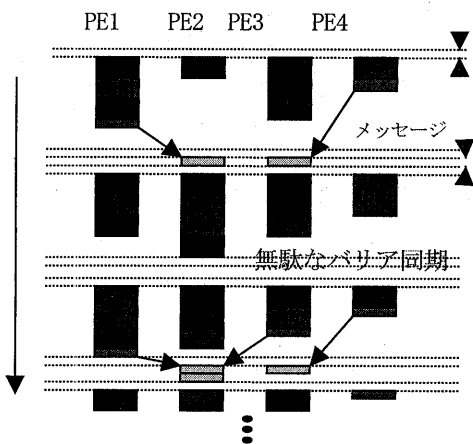


図1 同期法の無駄

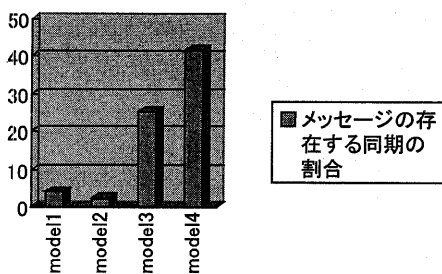


図2 有効なバリア同期の割合

効果

シミュレーション先読み同期法の効果を同期コスト、メッセージコスト、PEの稼働率の点から見る。

・同期コストの効果

バリア同期は各PEにメッセージを送りPE間での歩調を取るものである。よってそのコストが例えば最大50%減るという事は大変効果は大きい。

・メッセージコストの効果

同期が長くなるという事は、それまで小分けに1メッセージ、1メッセージずつ送っていたメッセージが2メッセージをまとめて送ることが出来るという事である。これまでの実験でメッセージ送信長が変わっても差ほど送信時間は変化しないことが分かっている。つまり複数のメッセージを1メッセージに梱包して送信するという事は、メッセージコストを半減させる。

・PEの稼働率

同期間隔が長くなる事で、これまでバリア同期までの各PEの事象処理の待ち時間が減少すると思われる。これにより各プロセッサの全体的な稼働率を向上させ、無駄のない並列計算が実現できる。

擬似乱数

シミュレーション先読みのための擬似乱数の選定を行う。シミュレーションにおける擬似乱数の定義は「なるべく予測不可能で周期の出来るだけ長いでたらめな数」という事になっており、これらを満たさなければならない。現在主要な擬似乱数発生器のベンチマーク(図3)をとり、そのなかで線形合同法(高速抽出法)を今回シミュレーション先読みのための擬似乱数とした。この乱数発生方式は次のようになる。

$$X_{(n+1)} = (X_n * p + q) \div r \quad (X_0 : \text{seed})$$

-qとrは互いに素な数

-p-1がrの素因数の倍数

-rが10のべき乗のとき、 $p \equiv 21 \pmod{200}$ 、かつ $q \equiv 5 \pmod{10}$

倍数ではないかつ q/r が 0.211 に近い値

この数式は再起計算式となっている。つまり前回の値 X_n がわかることにより、次の値 X_{n+1} が算出される事になる。よってこの X_n の値さえコントロールすれば、今後得たい乱数配列、過去に出力した乱数配列の再現も可能である。例えば、シミュレーション先読みを行う場合は現在の X_n の値を保存しておき、シミュレーション先読みを行う。シミュレーションを巻き戻すときは先に保存した X_n の値に戻す事で、同じシミュレーションをまた再現する事が出来る。これにより、戻したい地点の入力値さえ保存していれば、擬似乱数のコントロールを可能とする事が可能である。

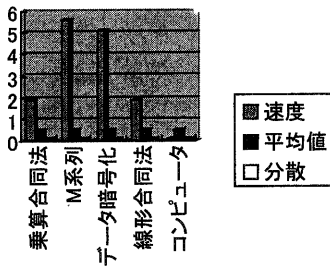


図3 擬似乱数のベンチマーク

環境

本研究における環境として、
 ホストプロセッサ
 MMX Pentium233MHz (メモリ 128MB) セルプロセッサ (4 台)
 Pentium II 450MHz (メモリ 384MB)
 によって構成された PC クラスタを用いた。なお、それぞれの PC はスイッチングハブを介して 100Base-TX によって接続されており、通信ライブラリとしては、現在、多くのプラットフォーム上で使用可能である MPI (Message Passing Interface) を用いた。対象モデルとしては 1024 ノードよりなる待ち行列システムシミュレータを用いた。

結果

シミュレーション先読み同期法の実行結果を載せる。モデル 1, 2 は比較的メッセージ数の少ないモデルであり、初期マップとしてランダムマップを用いたものがモデル 1、位置情報により均等に 4 分割されたマップを用いたものがモデル 2 である。モデル 3, 4 は比較的メッセージを発生しやすいモデルで、初期マップとしてランダムマップを用いたものがモデル 3、位置情報により均等に 4 分割されたマップを用いたものがモデル 4 である。それぞれのモデルについて、実行時間、総メッセージ数、同期回数、PE の稼働率についてグラフにした。

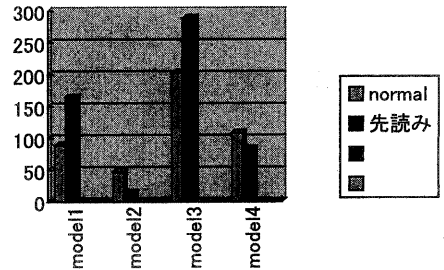


図4 先読み同期法の性能

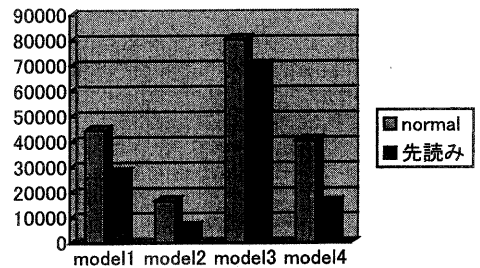


図5 メッセージ数の比較

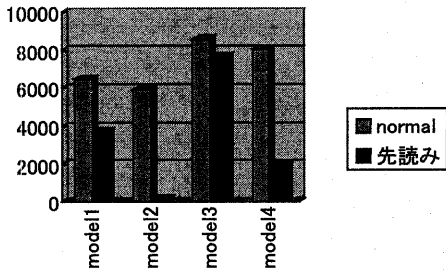


図6 同期コストの比較

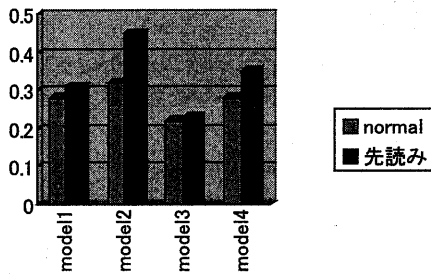


図7 PEの稼働率の平均比較

図6より同期回数が減少している事がわかる。しかしマッピングにランダムマッピングを比較した場合の同期回数の減少(model1, 3)は、面積でマッピングした場合(model2, 4)に比べ低い減少率となってしまった。ランダムマッピングは可能性的に隣接ノードがほぼ全て他のPEに属するノードという事が有りえ、トランザクションの移動が即メッセージとなってしまふ。そのため全体的にメッセージ送信回数が増加してしまい、メッセージ送信を含む同期の割合が増加してしまふ。(モデルによっては、90%以上というものも存在した) そのため、先読みシミュレーションを実施しても、メッセージに関する事象がすぐに出力されてしまい、大幅な同期コスト削減を実現できなかったと思われる。これより、シミュレーション先読み同期法の性能をより高めるためにはメッセージ送信回数を押

さえるマッピングを選択するのが好ましいと思われる。また、図5よりメッセージ送信回数が減少している事がわかる。これは、同期間隔を長く取れた結果、複数のメッセージ送信を一つのメッセージに梱包して送信できた結果だと思われる。しかしこれも同期回数の比較時と同様にランダムマッピングの際は減少率が低い。理由も同期回数のとときと同様である。稼働率のほうは、ランダムマッピングでさえも、やや同期間隔が長く取れたため、アイドル時間の低下が見られ上昇した。実行時間においてはランダムマップのほうは実行時間の上昇が見られた。これはシミュレーション先読みに関するコストが他のコスト削減に対応しきれなかった結果だと思われる。

先読みによる動的マッピング

バリア同期後すぐに先読みシミュレーションを行う上記のアルゴリズムでは、そこで各ノードでの事象処理の統計データを算出できる。そこで、突発的な事故などが発生した場合、それがGAでの計算時間では間に合わないほどの大きさであったときに、このデータを用いて動的マッピングを行う事でGAと組み合わせた更なる動的マッピングを目指す。統計データでマップ更新を行う際に使用した手法は面積マッピングである。その実行結果は次のようになった。

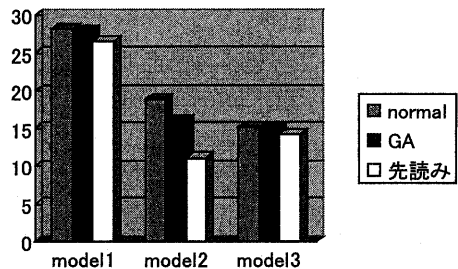


図8 先読みデータでの動的マッピング

図8を見ても分かる通り、GAだけでなく先読みデータからの動的マッピングを使用することによって更なる並列シミュレーションの実行効率の向上が見られた。これは未来に起こるシミュレーションデータを先読みして効率的なマッピングを作成できた結果だと思われる。

結論

今回、仮想時刻同期手法、マッピングの両点をシミュレーション先読みという観点から並列性を見出してきた。シミュレーション先読み同期法のほうは、ある程度効率のよいマッピングを使用しないと並列性は見出せないが、効果は十分なほど現れた。また、その効率的なマッピングに関しても、位置情報による面積マッピングなど比較的簡単なマッピングでも性能を出しており、有効な手法であると思われる。

統計データから算出されたマッピングに関しては、従来我々が研究してきたGAによる動的マッピングに組み合わせる事で、従来のGAの計算待ちの間の効率の悪いマッピングでのシミュレーションの問題を解決し、従来のもの以上の効果を得る事が出来た。

今回提案したシミュレーション先読みによる並列化は、先に起こるシミュレーションを前提として、並列性の向上を行っている。その結果、PDESの2つのアプローチにおいて効果を見出せた。このシミュレーション先読みという観点はPDESにおける新しい試みであり今後発展していくと思われる。

今後の展望

シミュレーション先読みはその他の手法、楽観的手法の分野でも有効であると思われる。また、統計データからの新たなマッピング作成に関しても、やや効率を悪化させる問題が発見されており、新たな手法を提案する必要がある。今後は、仮想時刻同期手法、マッピング手法のその他の分野においても、先読みアルゴリズムを適応させPDESの並列性を高めて行く

可能性は十分にある。

参考文献

- [1]根本貴由, 高井峰生, 成田誠之助, “ヌルメッセージ法を用いた待ち行列並列シミュレータの評価”, 信学技法, CPSY94-31, pp. 33-40, July, 1994.
- [2] RM Fujimoto, “Parallel and Distributed Discrete Event Simulation : Algorithms and Applications”, Proceedings of the 1993 Winter Simulation Conference, pp.106-114
- [3]三橋純、永田和敏、成田誠之助 “GAによる動的マッピングのPDESへの適用” 情報処理学会 2000
- [4]池野伸一・小山謙二, “現代暗号理論” 電子通信学会編 昭和61年9月1日発行
- [5] Richard M.Fujimoto, “Parallel Discrete Event Simulation”, Communication of the ACM, Vol.33, No.10, pp.30-53, October 1990.
- [5]森戸晋, 中野一夫, 相沢りえ子, “SLAM IIによるシステム・シミュレーション入門 改訂版”, 共立出版, 1993.
- [6]三橋純, “遺伝的アルゴリズムを用いた動的マッピング手法のPDESへの適用”, 2000年度修士論文