

離散事象並列シミュレーションにおける保守的同期手法の評価

高井 峰 生[†] 山城 登久二[†] 成田 誠之助[†]

離散事象並列シミュレーション (PDES) において、並列シミュレータの性能は適用する仮想時刻同期手法に大きく依存する。また、その仮想時刻同期手法の効率性はシミュレート対象モデルの性質、PDES 実行の前段階におけるシミュレート対象モデルの分割および並列計算機へのマッピングに大きく影響する。しかし、これらのことを考慮した仮想時刻同期手法の評価は行われておらず、従来の定性的な評価結果は手法の適用に有用でないことが多かった。本論文では、実際の PDES で生じる様々な状況を想定し、仮想時刻同期手法の定性的な比較・評価を行った。評価する仮想時刻同期手法は保守的手法の中で代表的な Chandy and Misra のヌルメッセージ法、問合せメッセージ法、受信統計に基づくヌルメッセージ法、バリア同期を用いる同期法の 4 手法とした。シミュレート対象モデルには基本的なトーラス結合待ち行列ネットワークを用い、並列計算機処理要素 (PE) の事象処理負荷の均衡、全体的な Lookahead 値の変化と局所的な Lookahead 値の変化、PE 間遷移事象数の増加など、モデルの性質やマップにともなう状況変化に対する性能測定を行った。その結果、実際問題に適用する仮想時刻同期手法を判断するために必要な各手法の特徴を把握することができた。

Evaluation of Conservative Synchronization Methods in Parallel Discrete Event Simulation

MINEO TAKAI,[†] TOKUJI YAMASHIRO[†] and SEINOSUKE NARITA[†]

In Parallel Discrete Event Simulation (PDES), the performance of a parallel simulator mostly depends on the virtual time synchronization method employed. The efficiencies of those methods are largely affected by the characteristics of the simulated model, the way of partitioning the simulated model and that of mapping them onto the parallel computer, which is a previous step of executing the PDES. However, little attention has been directed toward considering these conditions when these methods are evaluated. Thus the conventional results of qualitative evaluations have not yet been put into practical use. This paper compares and evaluates the virtual time synchronization methods qualitatively with due consideration given to various situations of PDES. Four conservative synchronization methods are selected and evaluated, i.e., Chandy and Misra's null message method, Query message method, Null message method based on receiving message statistics, and Synchronous method with barrier synchronization. Their performances are measured for a basic toroidal queuing network model with various conditions of parameters of the model and the ways of mapping such as balancing event processing load, changing global and local lookahead values, and increasing migrated events among PEs. The results indicate their features required for the choice of the method to be employed.

1. ま え が き

離散事象並列シミュレーション (Parallel Discrete Event Simulation 以下 PDES)^{1),2)} は、大規模な通信システム、計算機システムなどの設計・予備評価におけるシミュレート時間の短縮に有効である。近年に見られる離散事象システムの急激な大規模化・複雑化によって、PDES の必要性はますます大きくなってきているといえる。このような PDES を実用化するうえでの

問題点としては、並列計算機の処理要素 (Processing Element 以下 PE) 間で行われる仮想時刻 (シミュレート時刻) の同期方法^{1),3)} や、シミュレート対象モデルを分割し、各 PE にマッピングする方法^{4)~6)} などがあげられる。特に仮想時刻同期手法については従来から多くの研究がなされてきた。

仮想時刻同期手法の比較・評価は、特定のアプリケーションを想定したものと、想定しない定性的なものに分けられる。シミュレート対象システムの特性を利用しない定性的な比較・評価では、手法の一般的な特性が把握できる利点はあるが、反面、結果の有用性が比較・評価を行う条件に依存してしまう欠点がある。

[†] 早稲田大学理工学部

School of Science and Engineering, Waseda University

ここで、従来の仮想時刻同期手法の評価がどの程度実際問題に即した形で行われていたかを考えてみる。実際に PDES を行う場合、その前段階としてシミュレート対象モデルを分割し、各 PE へマッピングする作業がある。通常マッピングは各 PE の負荷分散、PE 間通信コストの最小化を目標に行われるが、特別な場合を除き、それぞれの目標に対する最適なマップは異なるため、実際の複雑なモデルでは両者に最適なマップを作成することはできない。そこで、各項目の変化への対応は PDES 実行時に適用する仮想時刻同期手法を選択するうえで重要である。しかし、従来はマッピングが比較的簡単なモデルに対して適切なマップを用いて手法評価が行われていたため、適切なマップを作ることが難しい実際問題と手法の評価条件との間に大きな隔たりがあった。

本論文では実際の PDES で生じる様々な状況を想定し、高並列計算機 AP1000⁷⁾を用いて仮想時刻同期手法の定性的な比較・評価を行う。評価する仮想時刻同期手法は保守的手法に絞り、その中で代表的な Chandy and Misra のヌルメッセージ法⁸⁾、問合せメッセージ法³⁾、受信統計に基づくヌルメッセージ法⁹⁾、実時間的に同期をとる同期法¹⁰⁾を評価対象とした。評価モデルには、多くの離散事象システムの基本となる待ち行列ネットワークを用い、モデルの性質やマップの変化にともなう各仮想時刻同期手法の実行性能変化を測定した。

本論文は 7 章から構成される。2 章では PDES の基本原理と比較・評価した仮想時刻同期手法のアルゴリズムについて述べる。3 章では各シミュレータの実装法について述べ、4 章で評価モデルについて説明する。5 章で評価および考察を行い、その結果から各仮想時刻同期手法の特徴を 6 章でまとめる。

2. 仮想時刻同期手法

2.1 PDES の基本アルゴリズム

離散事象シミュレーションは、システム内に生じる事象を生起時刻順に処理してシミュレーションを進める。PDES では、さらにそれらの事象を生起した場所により分類して各 PE に割り当て、並列処理を行う。

各 PE は事象処理中に他の PE で処理すべき事象が発生した場合、それをメッセージとして送出する。送信 PE は、受信 PE に対して今後ある仮想時刻までの事象を送出しないことが保証できる場合、その仮想時刻（保証時刻）をメッセージに時刻印として付加する。各 PE は、受信メッセージの時刻印までに生起した事象だけを処理することで、割り当てられた場所に生じ

る事象すべてを生起時刻順に処理できる。ただし、事象を含むメッセージの情報だけではデッドロックを生じてしまう可能性があるため、事象を含むメッセージの送出以外にも仮想時刻の同期をとる必要がある。

この仮想時刻同期を行う手法は、大きく保守的なものと楽観的なものとに分類される¹⁾が、不当な事象処理を一時的に容認する楽観的手法は、その性能がモデルの状態保存コストに大きく依存するため定性的な評価が難しい。また、楽観的手法と保守的手法とでは評価すべき点が大きく異なるため、両者の手法をまとめて比較・評価することは困難である。本論文では、保守的な仮想時刻同期手法に絞って議論を進める。

2.2 ヌルメッセージ法 (NULL)

Chandy and Misra のヌルメッセージ法⁸⁾は、仮想時刻同期手法の中で最も古典的かつ基本的な手法である。各 PE は他の PE に対する保証時刻が更新できる場合、その PE に送出する事象がなくても空のメッセージ（ヌルメッセージ）を送出する。これによって、他の PE からのメッセージを待つことから生じるデッドロックを防ぐことができる。

従来の研究から、ヌルメッセージ法は大量のヌルメッセージを送出し、大きな通信コストを必要とすることが指摘されている。

2.3 問合せメッセージ法 (DMND)

問合せメッセージ法³⁾はヌルメッセージ法における不必要なヌルメッセージ送出を抑えるために提案された手法である。問合せメッセージ法では、他の PE への保証時刻が更新できてもヌルメッセージを送出しない。その代わりに、メッセージ受信 PE は送信 PE からの保証時刻更新を必要とする場合、当該 PE へ問合せメッセージを送出する。メッセージ送信 PE は、問合せメッセージを受信したときに限ってヌルメッセージを送出する。

このように、問合せメッセージ法ではヌルメッセージ送出を要求駆動にすることにより、不必要なヌルメッセージ送出を抑えている。ただし、問合せメッセージを送出してからヌルメッセージを受け取るまで、保証時刻を問い合わせた PE は事象を処理できないので、メッセージ待ち時間はヌルメッセージ法よりも長くなることが予想される。

2.4 受信統計に基づくヌルメッセージ法 (STAT)

受信統計に基づくヌルメッセージ法⁹⁾では、PE_i は PE_j へ送出する事象がある場合、s_j 個の事象を処理す

★ 正確には後述する Lookahead 値の和が 0 になるループが存在するとデッドロックを生じるが、その議論は本論文の内容に直接関係ないので省略した。

ごとにメッセージを送出する。PE_j は PE_i から受信した総メッセージ数 r_i と、受信時の状態が PE_i からのメッセージ待ちであった回数 r_i^w を記録しておき、 r_i^w/r_i が大きいときは PE_i に対してメッセージ送出頻度の増加要求メッセージを送出する。この増加要求メッセージを受信した PE_i は、 $s'_j = \alpha_j s_j$ ($\alpha_j < 1.0$) とし、以後のメッセージ送出行う。同一の PE からメッセージ送出頻度の増加要求を受けたときには、さらに $s'_j = \alpha_j s_j$ として s_j を小さくしていく。すべての PE でメッセージ送出頻度増加要求が発生しなくなったら、設定されている s_j を以後のメッセージ送出間隔とする。

受信統計に基づくヌルメッセージ法では、このように各 PE のメッセージ受信統計に基づき、PE のメッセージ待ち時間を増加させない範囲で送出メッセージ数の削減を図る。

2.5 同期法 (SYNC)

上記3手法は各 PE が仮想時刻同期を非同期に行う手法であった。これに対し、同期法¹⁰⁾では仮想時刻同期の際にバリア同期を行い、他のすべての PE に対する保証時刻を同時に計算する。各 PE は、計算されたその保証時刻までの事象を他の PE に関係なく処理できる。このため、送出メッセージには保証時刻を知らせるための時刻印を付ける必要がなく、他の3手法に比べて保証時刻の計算回数は少なく済む。

3. 並列シミュレータの実装

本章では、並列シミュレータの性能を左右する事象リストの構造、Lookahead 値の計算、メッセージ送受信の効率化について、実装上の工夫点を述べる。ただし、以後ヌルメッセージ法を NULL、問合せメッセージ法を DMND、受信統計に基づくヌルメッセージ法を STAT、同期法を SYNC と略記することにする。

3.1 事象リストの構造

これから処理する事象を生起時刻順に登録しておく事象リストは、離散事象シミュレータの中心的存在であり、事象リストを扱う処理は全体の処理の大半を占めている。この事象リストには通常連結リスト構造を用いるが、大規模なモデルのシミュレーションでは事象リストが長くなり、リスト操作のコストは非常に大きくなる。そこで、本論文ではリストが長い場合に有利なヒープ構造を事象リストに採用した。ただし、ヒープ構造では同時刻に生起した事象が複数あると、それらの事象の取出し順序が事象リストに登録した順序と必ずしも同一にならない。同時刻に生起した事象は事象リストに登録された順序で処理する必要がある

ため、本シミュレータでは各事象に登録順序を示すシリアル番号を付加し、同時刻に生起した事象の登録順序と取出し順序とを同一にする工夫をした。

3.2 Lookahead 値の計算

PE が現在シミュレートしている仮想時刻と、その PE のメッセージ送出先 PE への保証時刻との差を Lookahead 値と呼ぶ。この Lookahead 値の大きさは仮想時刻同期手法の効率に大きく影響してくる^{1),11)}ので、性能向上のためにはなるべく大きな Lookahead 値を計算する必要がある。ただし、1個の PE に割り当てられてたモデルが大きいと、メッセージ送出のたびに事象リストの内容を調べ、送出先 PE への最大保証時刻を計算するオーバーヘッドは無視できなくなる。そこで、本シミュレータでは事象リストの先頭事象の生起時刻 t に、サービス終了後 PE_i へトランザクションが移動する可能性のあるサーバの最小サービス時間 d_i を加えた $t + d_i$ を PE_i への保証時刻とした。

3.3 メッセージ送受信の効率化

• NULL

基本的な NULL のアルゴリズムを述べた文献⁸⁾と同様に仮想時刻同期を行うと、各 PE はヌルメッセージを受信して他の PE への保証時刻が更新できれば、後から他のメッセージが到着していてもヌルメッセージを送出する。これは、すでに同一 PE からより大きな時刻印のメッセージが到着している場合、無意味なメッセージ送出となる。

そこで、メッセージ受信は事象処理が行えなくなった時点でを行い、それまでに到着しているメッセージを一括して取り込んで、受信メッセージに含まれるヌルメッセージを除去した。これによって、不必要なヌルメッセージの伝搬・生成を抑えた。

• DMND

DMND はヌルメッセージを要求駆動にしてメッセージの大量発生を防ぐ手法であるが、事象を含むメッセージ送出に関しては NULL の送出法と同一である。そこで、問合せメッセージを受信するまでは事象を含むメッセージも送出しないことにし、送出する事象をとりまとめてメッセージ送出の効率化を図った。

• STAT

STAT では、適切な r^w/r の閾値および α の値を設定する必要がある。本論文では、並列計算機を用いていくつかのモデルをシミュレートした際の経験値に基づき、受信統計処理を短時間で終了させるために $r^w/r \leq 0.1$, $\alpha = 0.5$ とした。

• SYNC

SYNC では、事象処理中に他の PE で処理する事象が

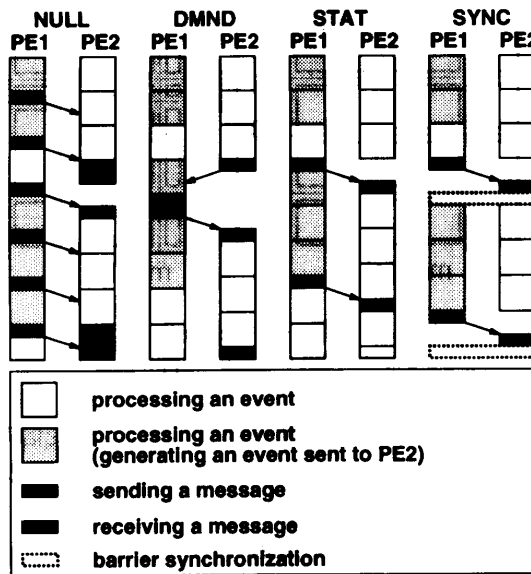


図1 各手法による PDES の進行例
Fig. 1 Progress of a PDES with each method.

発生しても、その事象は必ずバリア同期を1回以上行った後に処理される。そこで、他のPEで処理される事象をバリア同期を行う直前にまとめて送出して、メッセージ送出の効率化を図った。

以上のようなメッセージ送受信効率化後の各シミュレータの処理進行例を図1に示す。図1では、PE₁はPE₂へ送出する事象が5個発生し、PE₂はシミュレーション開始から3事象処理したときにPE₁が計算する3事象処理後の保証時刻を必要としている。ただし、STATではPE₁の s_2 が3に設定されており、SYNCではシミュレーション開始時点で計算した保証時刻によって両PEともに3事象処理できるとする。

4. シミュレート対象モデル

本章では、仮想時刻同期手法の比較・評価に適切なシミュレート対象モデルについて考える。評価モデルに複雑なものを用いると、実行時性能を決定する要因の解析が複雑となり、各手法の特徴を見出すことが困難になる。しかし、単純なモデルと適切なマップを用いれば実際問題との差が大きくなり、有用な結果は得られなくなる。そこで、本論文では結合関係が単純な二次元トラス結合の待ち行列ネットワークモデルを用い、モデルのパラメータを変化させたり、様々なマップを当てはめることにより、実際問題で生じる状況を作り出して評価を行うことにした。二次元トラス結合の例を図2に示す。

トラス結合(メッシュ結合)自体は文献11)や12)の評価にも用いられているが、通信網や並列計算機の

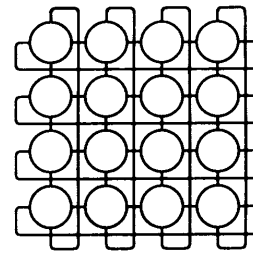


図2 二次元トラス結合の例
Fig. 2 An example of 2-D toroidal connectivities.

相互結合網などに使われている実際的な結合形態である。また、結合されたノード間の有向ループ数および任意の二点間経路数が多いため、仮想時刻同期手法の性能差が大きく出ること期待できる。本論文では、PDESを要求しているシステムが大規模であること、多くの分割方法が選べることなどから、 $32 \times 32 = 1024$ 個の待ち行列サーバを結合させた二次元トラスネットワークを作成して評価に用いた。

シミュレート対象モデルには、待ち行列サーバ間の結合以外にもいくつかのパラメータを設定する必要がある。まず、システム内を動きまわるトランザクション数は、すべての初期待ち行列長を3として総計3072個に設定した。また、各サーバはFCFS(First Come First Served)の順序でサービスするトランザクションを決め、 $[d, 2.0 - d][ut]$, $0 \leq d \leq 1.0$ (ただし[ut]は仮想単位時間)の一様分布に従う時間のサービスを行うことにした。これにより、各サーバの平均サービス時間は d に関わらず $1.0[ut]$ となり、最小サービス時間は d によって変化させることができる。

さらに、トランザクションが各サーバのサービス終了後に隣接サーバに移動する確率は均等(1/4)にした。この確率が均等であれば、各待ち行列サーバにはほぼ同数のトランザクションが到着するため、各待ち行列サーバをシミュレートするコストが同じになる。これによって、全体に対して大きなシミュレートコストを持つ分割不可能な部分がなくなり、適用した仮想時刻同期手法と無関係にシミュレート時間が決まってしまうことは避けられる。

この評価モデルは均等な分岐確率と平均サービス時間を持っているため、シミュレーションが定常的になることを保証できる。

5. 評価

本章では、前章で述べたシミュレート対象モデルのパラメータとマッピングの変化で実際問題に生じる状況を作り出し、各手法の特性を把握する。

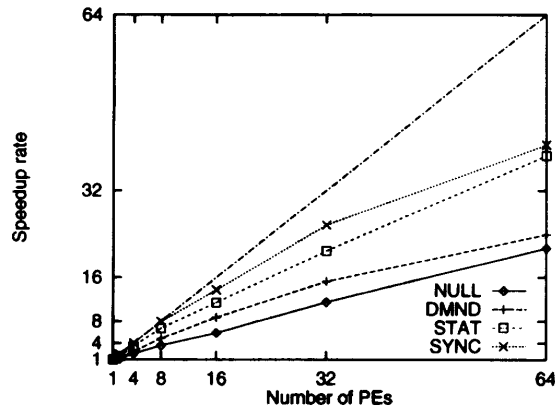


図3 PE数に対する速度向上率

Fig. 3 Speedup rate against the number of PEs.

表1 NULLを基準とした送出メッセージ数 (16 PE 利用時)
Table 1 Number of sent messages normalized by the NULL's number (utilizing 16 PEs).

NULL	DMND	STAT	SYNC
1.00000	0.08171	0.14025	0.01792

5.1 標準的なマップによる実行

まず、いろいろな条件でシミュレートする前に、基準となる標準的なマップ (モデルを PE 数に合わせて均等に分割したマップ) を用いたシミュレーションを行った。その際の速度向上率を図 3 に示す。ただし、サーバの最小サービス時間はすべて $d = 0.5$ にしてある。

図 3 の速度向上率は NULL, DMND, STAT, SYNC の順に大きくなっており、NULL は速度向上率が SYNC の約半分と効率が悪い。この場合、利用 PE 数が同一なら各手法の事象処理コストは同じであるので、速度向上率の差は通信コストの大きさに関係していると考えられる。また、PDES では送出メッセージの大きさが小さいため、通信コストは送出メッセージ数に比例していると見なせる。

そこで、16 PE 利用時に各手法で送出されたメッセージ数を表 1 に示す。ただし、表 1 の数値は NULL の送出メッセージ数によって正規化してある。表 1 から、NULL は他手法に比べて非常にメッセージ数が多く、このことが速度向上率の低下につながっているといえる。しかし、送出メッセージ数は NULL, STAT, DMND, SYNC の順に少なくなっているが、この順番と速度向上率の大きさの順番は同一でなかった。DMND のメッセージ送出数は少ないが、2.3 節で述べたように他手法よりもメッセージ待ち時間が長くなるため、メッセージ送出数の多い STAT の方が DMND よりも高速に

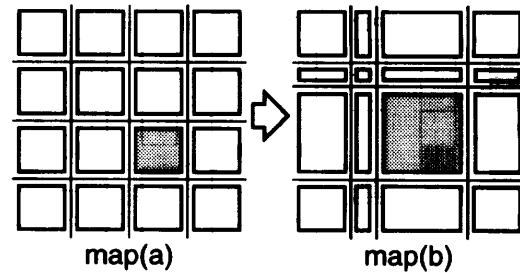


図4 負荷の不均等なマッピング
Fig. 4 Unbalanced mapping.

なっている。また、SYNC ではメッセージ送受信のほかにバリア同期も行っているため、送出メッセージ数が最も少ないから効率的であるとはいえない。

5.2 事象処理負荷の変動

効率的な PDES を行ううえで配慮しなければならないのは、各 PE の負荷を均等にし、PE 間の通信コストを最小にする^{4)~6)}ことである。各 PE の負荷は、仮想時刻同期を除けば主に事象処理によるものであるから、負荷分散は各 PE の処理事象数を均等化することにあたる。しかし、実際にはモデル内に発生する事象数の分布が予測できなかつたり、予測できても処理事象数の多い部分の分割が不可能であったりして、処理事象数の均等化は難しい。さらに、処理事象数の均等分割が可能であっても、あえてそれを犠牲にして通信コストを小さくする場合もあるので、負荷が不均等なときの仮想時刻同期手法の性能は重要である。

本論文で用いたシミュレート対象モデルでは、待ち行列サーバという単位で計算量の予測がつく。したがって、二次元トラス結合モデルを図 4 の map(a) のように均等分割、マッピングをすることで、各 PE の処理事象数はほぼ同数になり、負荷も均等化される。しかし、前述の理由による不均等な負荷への対応を調べるため、図 4 の map(b) のように不均等に分割した場合の実行時間を測定した。その結果が図 5 である。ここで、横軸は図 4 の map(b) における網掛け部分のサーバ数を map(a) における網掛け部分のサーバ数 (64 個) で割った値であり、縦軸は map(a) による実行時間によって正規化してある。

全体の実行時間は処理事象数が最も多い PE の実行時間に依存するため、図 5 では単純な比例関係が成り立つはずであり、実際に DMND を除く 3 手法ではこの関係が成り立っているといえる。DMND は、map(a) を用いた場合の速度向上率が 8.86 であるから、1 PE の最大サーバ数が標準のマップの 2.25 倍を超えた時点で逐次シミュレーションよりも遅くなっている。

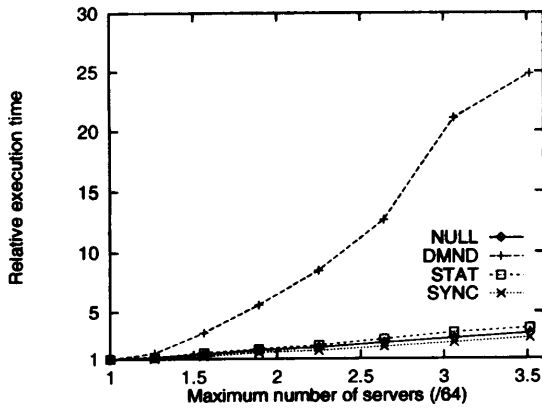


図5 事象処理負荷の変動に対する実行時間

Fig. 5 Execution time against event processing load.

図4 map(b)の網掛け部分を割り当てられたPEでは、単位仮想時間あたりの処理事象数が多くなるため、他のPEよりシミュレートしている仮想時刻が遅れることになる。DMNDでは、このようなPEは他のPEから多くの問合せメッセージを受信することになるので、メッセージ送受信数も多くなってしまふ。つまり、事象処理コストの最も大きなPEが通信コストも最大になるので、比例関係以上の実行時間増加が生じたものと思われる。

この結果からDMNDは、負荷が不均等な場合への対応が不適切であり、シミュレート対象モデルが均等に分割できなかつたり、各PEの計算能力が大きく異なる場合には利用すべきでないといえる。

5.3 Lookahead値の変化

3.2節でも述べたが、従来の研究からLookahead値は通信コストに大きな影響を与えることが分っている。Lookahead値はシミュレート対象モデルの特徴によって変化するが、それに加えてモデルの分割・マッピング方法によっても変化する。実際のシミュレート対象モデルでは、全体的にLookahead値を大きくするマッピングを試みても局所的に小さなLookahead値を持ってしまう場合があるので、局所的なLookahead値変化への対応も手法選択のうえで重要である。そこで、本論文では全体的なLookahead値変化に加えて局所的なLookahead値変化についても調べる。

5.3.1 Lookahead値の全体的な変化

16 PEを用い、すべてのサーバの最小サービス時間を同時に $d = 0.01$ から $d = 0.95$ まで変化させたときの速度向上率を図6に示した。図6から、すべての手法において最小サービス時間の減少にともなう性能劣化が見られる。

性能劣化の主な原因は、1回の保証時刻更新によって

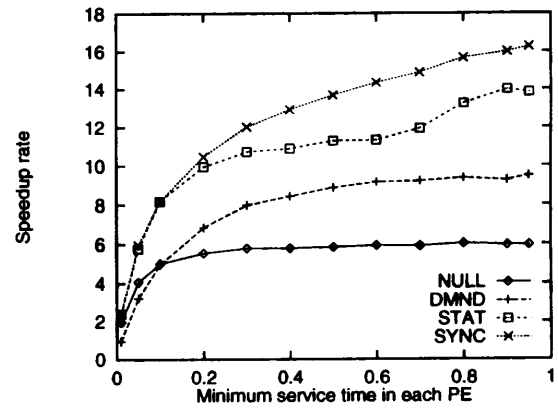


図6 全体的な最小サービス時間に対する速度向上率

Fig. 6 Speedup rate against the minimum service time of all the servers.

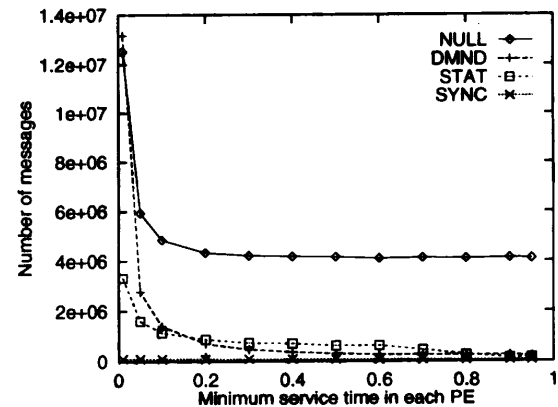


図7 全体的な最小サービス時間に対する送出メッセージ数

Fig. 7 Number of sent messages against the minimum service time of all the servers.

処理できる事象数がLookahead値の減少とともに少なくなり、メッセージ送出数(SYNCではバリア同期回数)の増加を招くためと考えられる。そこで、サーバの最小サービス時間値と送出メッセージ数との関係を図7に示した。図7から、SYNCを除く3手法の送出メッセージ数は最小サービス時間値とほぼ反比例の関係になっていることが分かる。SYNCは、メッセージ送信数は $d = 0.95$ から $d = 0.01$ になっても1.8倍程度にしかなっていないが、バリア同期回数は84.7倍になっており、他の3手法と同様 d の値に大きな影響を受けている。

図6で最も性能劣化の激しかったDMNDは、1回の保証時刻更新を行うために2つのメッセージが必要であるため、図7でも他の手法より送出メッセージ数の

☆ 保証時刻の計算では、事象リストの先頭事象の生起時刻を d に加算するので、Lookahead値は d よりも大きくなる。このため、各曲線は反比例曲線を x 軸の負の方向に移動した形になっている。

増加が激しい。このため、全体的に小さいLookahead値しか得られない場合はDMNDを適用しない方がよいといえる。

5.3.2 Lookahead 値の局所的な変化

Lookahead 値の局所的な変化に対する各手法の対応を調べるために、図 8 の網掛け位置のサーバの d だけを変化させた場合を考える。図 8 の網掛け位置のサーバでは、サービス終了後に PE_0 または PE_5 へトランザクション（事象）が移動する可能性がある。よって、このサーバの d 値を変化させると、 PE_4 が計算する PE_0 と PE_5 への Lookahead 値を変化させることになる。この網掛け位置のサーバを $d = 0.0$ から $d = 0.95$ まで変化させ、他のサーバは $d = 0.95$ に固定した場合の速度向上率を図 9 に示す。

図 9 で、非同期な仮想時刻同期を行う 3 手法は局所的な d 値変化の影響を受けていないが、SYNC は図 6 の場合と同様な影響を受けている。SYNC では局所的に Lookahead 値が小さくても、バリア同期時に計算するすべての PE に対する保証時刻値が小さくなり、バリア同期回数が増加するのが原因である。この

ため、部分的に Lookahead 値が小さくなることの避けられない状況では、SYNC は選択すべきでないといえる。

5.4 PE 間遷移事象数の変化

仮想時刻同期を含めた各 PE の通信コストはシミュレーション開始前に予測できないので、従来は PE 間遷移事象数が送出メッセージ数と一致すると仮定し、通信コストの指標として扱ってきた。そこで、この PE 間遷移事象数と仮想時刻同期手法の効率との関連を調べてみる。

トラス結合モデルを 4 個の PE にマッピングする場合、通常は図 10 の map(c) のようにマッピングするが、モデルを 16 分割して map(d) のようにマッピングすると PE 間遷移事象数は map(c) の 2 倍になる。ただし、図中の数字は分割されたモデルを割り当てる PE 番号を表す。このように、利用 PE 数よりも多くの数にモデルを分割し、各 PE に不連続に割り当ててシミュレーションを行ったときの実行時間を図 11 に示す。ただし、図 11 の横軸は標準的なマップによる実行で生じた PE 間遷移事象数によって正規化してある。

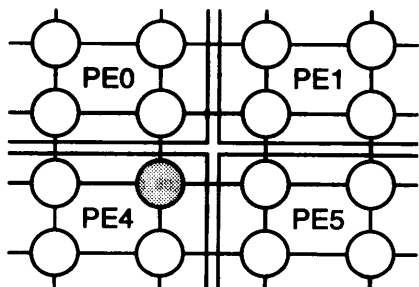


図 8 最小サービス時間が変化するサーバの位置
Fig. 8 Location of the server which changes its minimum service time.

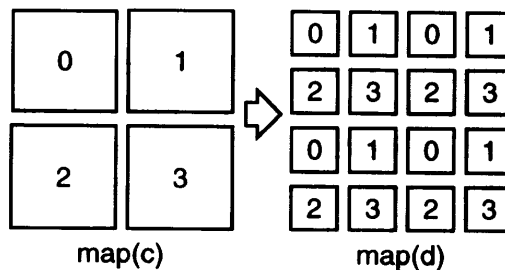


図 10 マッピングによる PE 間遷移事象数の増加
Fig. 10 An increase of the number of migrated events among PEs caused by difference in mapping.

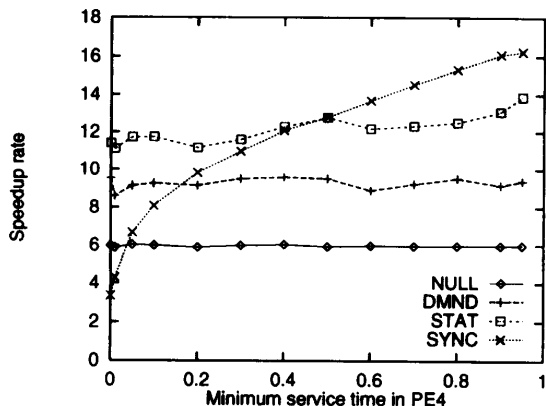


図 9 PE_4 上のサーバの最小サービス時間に対する速度向上率
Fig. 9 Speedup rate against the minimum service time of the server in PE_4 .

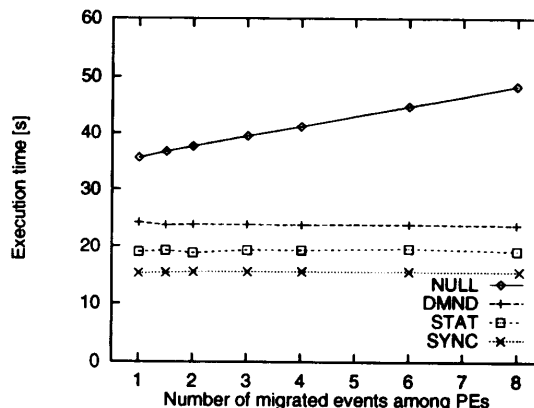


図 11 PE 間遷移事象数に対する実行時間
Fig. 11 Execution time against the number of migrated events among PEs.

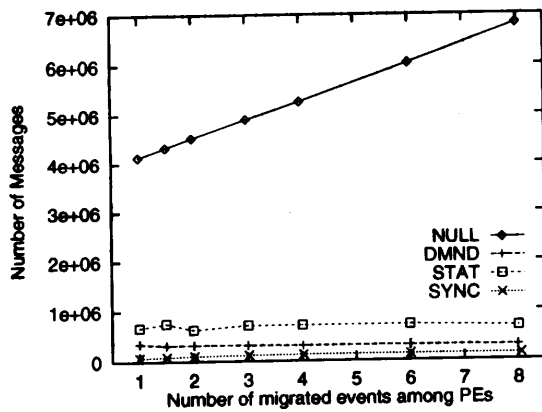


図12 PE間遷移事象数に対する送出メッセージ数
Fig. 12 Number of sent messages against the number of migrated events among PEs.

図11では、NULLを除いてPE間遷移事象数による実行時間変化は見られない。このときの送出メッセージ数を示した図12でも、NULL以外の送出メッセージ数に変化はない。また、増加の傾向にあるNULLでも、PE間遷移事象数が約8倍になったときの送出メッセージ数は標準的なマップを用いた場合の1.65倍であり、従来仮定していたPE間遷移事象数と送出メッセージ数との一致は成り立っていない。そこで、各仮想時刻同期手法の送出メッセージ数を決定する要因について考えてみる。

まず、NULLでは事象の遷移がないPEにも保証時刻の更新があればヌルメッセージを送出する。そのため、PE間遷移事象数が増えても事象を含むメッセージ数が増えるだけで、ヌルメッセージも含めた総メッセージ数は変化しないと考えられる。ただし、すでにヌルメッセージによって保証時刻が更新されているPEへ送出する事象が発生した場合、同一保証時刻を持つメッセージが再送出されるため、図12ではPE間遷移事象数にともなって送出メッセージ数が増加したのと思われる。DMNDやSTATでは、メッセージ送受信数は送出する事象の発生数よりも、むしろメッセージ受信PEが保証時刻更新を必要とする回数に依存する。よって、メッセージ送出先PE数が同じであれば送出メッセージ数は変化しないものと考えられる。また、SYNCでは送出メッセージ数がバリア同期回数を超えないことは3.3節のメッセージ送受信効率化から明らかである。バリア同期回数はLookahead値に反比例するが、PE間遷移事象数には無関係なため、SYNCの送出メッセージ数もPE間遷移事象数と無関係であるといえる。

仮想時刻同期手法を用いずメッセージ送出の効率化も図らない場合、PE間遷移事象数と送受信メッセー

ジ数は正確に一致し、通信コストも比例すると考えられる。しかし、実際は仮想時刻同期手法が用いられるうにメッセージ送受信の効率化も試みられるので、PE間遷移事象数と通信コストとの関係は弱くなってしまう。Lookahead値にともなう送出メッセージ数変化を示した図7と、PE間遷移事象数にともなう変化を示した図12の縦軸のスケールなどを比較すると、PE間遷移事象数は今回評価した他の状況に比べ、マッピングの際にも仮想時刻同期手法を選択する際にも重要でないといえる。

6. まとめ

以上の評価結果から各手法の特徴を整理すると次のようになる。

• NULL

NULLは、5.1節のような単純な状況においてもメッセージ送出数が多く、全体的に効率の悪い手法であった。ただし、NULLのメッセージ送出効率化を図ったDMNDは、場合によってはNULLより性能が劣化しており、メッセージ送受信効率化の方法には十分注意しなければならない。

• DMND

DMNDは、バリア同期を用いるSYNCを除けば最も送受信メッセージ数が少なく、本来の目的であるNULLの送受信メッセージ数削減には成功している。ただし、全体的にLookahead値が小さいときや負荷が均等でない状況では、NULLよりも送受信メッセージ数が多くなって性能劣化が大きい。このため、通常はNULLのメッセージ送受信を異なる方針で効率化したSTATを用いた方がよいと思われる。しかし、たとえばマルチタスク環境ではSTATの利用する受信統計は意味を持たないので、STATを適用することはできない。そのような環境でもDMNDは適用できるため、Lookahead値や負荷分散に対する難点を回避できればDMNDを選択することもあるといえる。

• STAT

STATは今回の測定結果を示すいくつかの図で不自然な曲線を描いている。この原因は、STATの受信統計処理を短時間で終了させたため、設定された s 値が最適値より小さくなってしまい、メッセージ送受信効率が悪化したためと考えられる。これは、 s 値の設定がSTATの性能に大きな影響を与えることを意味している。しかし、STATは最大速度の点でSYNCには及ばなかったものの、今回試験した負荷変動、Lookahead値変化に対して最も安定した性能を発揮した。このことから、STATは実際問題における複雑な状況では今

回評価した手法の中で最も有効であるといえる。ただし、先にも述べたが、受信統計が意味をなさない状況ではSTATは利用できないので注意が必要である。

• SYNC

SYNCは全体を通して効率の良い手法であったが、バリア同期を用いているため局所的なLookahead値変化に対して性能変化が大きかった。しかし、この局所的なLookahead値に対する難点をマッピングなどで補えば、今回比較・評価した手法の中で最も効率の良いPDESを行うことができる。また、SYNCはマルチタスク環境など、バリア同期コストが大きくなる状況では性能を発揮しないことが予想される。これらのことから、SYNCの選択にはマッピング、計算機環境などを特によく考える必要があるといえる。

7. む す び

本論文では、実際問題でPDESを行う際に生じる状況を想定し、仮想時刻同期手法の評価を行った。その結果、適用する手法を判断するために必要な各手法の特徴を把握することができた。本論文で想定した仮想時刻同期手法に好ましくない状況は、場合によってはPDES実行の前段階におけるマッピングで補えることもある。そこで、今後はマッピングの段階までを含め、全体として効率の良いPDESの研究を行っていききたい。

謝辞 本研究でAP1000を利用するにあたり、貴重なご意見をいただいた富士通研究所並列処理研究センターの方々に深く感謝いたします。

参 考 文 献

- 1) Fujimoto, R.M.: Parallel Discrete Event Simulation, *Comm. ACM*, Vol.33, No.10, pp.30-53 (1990).
- 2) Fujimoto, R.M.: Parallel and Distributed Discrete Event Simulation: Algorithms and Applications, *1993 Winter Simulation Conference*, pp.106-114 (1993).
- 3) Misra, J.: Distributed Discrete-event Simulation, *ACM Computing Surveys*, Vol.18, No.1, pp.39-65 (1986).
- 4) Nandy, B. and Loucks, W.M.: An Algorithm for Partitioning and Mapping Conservative Parallel Simulation onto Multicomputers, *6th Workshop on Parallel and Distributed Simulation*, pp.139-146 (1992).
- 5) Fujimoto, R.M. and Nicol, D.M.: State of the Art in Parallel Simulation, *1992 Winter Simulation Conference*, pp.246-254 (1992).
- 6) Takai, M., Nemoto, T., Abe, Y. and Narita,

S.: A Partitioning and Mapping Method for Parallel Simulation of Large Scale Queuing Networks, *The 1st International Workshop on Simulation in Manufacturing Systems*, AIT, pp.54-61 (1994).

- 7) Shimizu, T., Ishihata, H. and Horie, T.: Low-Latency Message Communication Support for the AP1000, *The 19th International Symposium on Computer Architecture*, pp.288-297 (1992).
- 8) Chandy, K.M. and Misra, J.: Distributed Simulation: A Case Study in Design and Verification of Distributed Programs, *IEEE Trans. Softw. Eng.*, Vol.5, No.5, pp.440-452 (1979).
- 9) 高井峰生, 山城登久二, 成田誠之助: 離散事象並列シミュレーションにおける効率的なメッセージ送出則, *情報処理学会論文誌*, Vol.37, No.3, pp.430-438 (1996).
- 10) Nicol, D.M.: The Cost of Conservative Synchronization in Parallel Discrete Event Simulations, *J. ACM*, Vol.40, No.2, pp.304-333 (1993).
- 11) Fujimoto, R.M.: Performance Measurements of Distributed Simulation Strategies, *Trans. of the SCS*, Vol.6, No.2, pp.89-132 (1989).
- 12) Lubachevsky, B.D.: Efficient Distributed Event-Driven Simulations of Multiple-Loop Networks, *Comm. ACM*, Vol.32, No.1, pp.111-123 (1989).

(平成7年10月26日受付)

(平成8年9月12日採録)



高井 峰生 (学生会員)

平成4年早稲田大学理工学部電気工学科卒業。平成6年同大学院理工学研究科電気工学専攻修士課程修了。現在同大学院博士課程在学中。平成7年同大学情報科学研究教育センター助手。平成8年同大学メディアネットワークセンター助手。各種アプリケーションの並列処理に関する研究に従事。電子情報通信学会, 日本オペレーションズ・リサーチ学会, SCS各会員。



山城登久二 (学生会員)

平成7年早稲田大学理工学部電気工学科卒業。現在同大学院理工学研究科電気工学専攻修士課程在学中。各種アプリケーションの並列処理に関する研究に従事。

**成田誠之助**

昭和 37 年早稲田大学大学院修士課程修了。昭和 37 年アメリカ・バドュー大学大学院留学（フルブライト留学生）。昭和 38 年早稲田大学理工学部助手，以後，講師・助教授を経て，昭和 48 年教授，現在に至る。工学博士。分散計算機制御システム，並列処理，産業用ロボット制御，デジタル制御理論，CIM などの研究に従事。計測自動制御学会，電気学会，ロボット学会，IEEE 各会員。
