

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付則

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) に関する多くの策は、シミュレーション実行中に小さなメッセージを多数送付する必要があり、密結合の並列計算機を用いない限り大きな性能劣化を引き起こす可能性がある。このことは、並列シミュレータを構成する各処理要素に割り当てる処理コストが比較的小さな場合避けることができないが、実際に PDES を必要とする状況ではメッセージの取りまとめが可能である。しかし、従来は全体のシミュレーションコストと各処理要素への割り当てコストとの比を考慮していなかったため、メッセージ送付効率化の研究はほとんどなされていなかった。本論文では、PDES における効率的なメッセージ送付則 EMSR (Efficient Message Sending Rule) を提案する。この送付則は仮想時刻同期の観点から、各処理要素のメッセージ待ち時間を増やすことなく不必要なメッセージの削減を行うことを目的とする。このメッセージ送付則を評価するため、Chandy and Misra のヌルメッセージ法におけるメッセージ送付を改良し、確率的に EMSR に従う待ち行列シミュレータを並列計算機 AP1000 上に実現した。従来のメッセージ送付と比較した結果、EMSR に従うことによって、より少ないメッセージ数とメッセージ待ち時間でシミュレーションを行うことができた。さらに、メッセージ通信速度による性能劣化が従来のメッセージ送付より小さく抑えられることも分かった。

An Efficient Message Sending Rule in Parallel Discrete Event Simulation

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

Many schemes in Parallel Discrete Event Simulation (PDES) generate a lot of tiny messages in the simulation, which causes considerable performance degradation unless the processing elements are closely coupled. This problem is not avoidable when cost assigned to each processing element is small compared with the total cost for the simulation, but those tiny messages can be gathered in actual cases. Little has been studied for efficient message handling because the fraction of the cost assigned to each processing element is not considered. This paper proposes EMSR (Efficient Message Sending Rule) for efficient communication in PDES. This rule aims at reducing unnecessary messages for virtual time synchronization without increasing the message waiting time of processing elements. To evaluate the proposed rule, the message sending style of Chandy and Misra's null message method is tuned up so that each processing element obeys EMSR stochastically, and a queuing network simulator with this message sending style is implemented on a parallel computer, AP1000. This simulator is compared with the simulators with two conventional message sending styles. It is found that EMSR requires a smaller number of messages and time for waiting messages. Furthermore, performance degradation caused by communication speed is smaller than the conventional ones.

1. ま え が き

システムの状態変化が離散的に生じると見られるシステムを総称して離散事象システムと呼ぶ。近年、通信分野の ATM 技術に代表される高速ネットワークの普及や計算機の機能向上など、離散事象システムの大

規模化、複雑化には目を見張るものがある。これにともない、離散事象システムの設計、および評価に用いられるシミュレーションも多大な計算コストを要求されるようになった。このような背景から、高速化への試みとして離散事象並列シミュレーション (Parallel Discrete Event Simulation, 以下 PDES)^{1),2)}の研究が多くなされている。

PDES では、シミュレート対象システムを状態変数の独立したいくつかの物理プロセス (physical pro-

[†] 早稲田大学理工学部

School of Science and Engineering, Waseda University

cess, 以下 pp) のネットワークとして表現し, それぞれの pp をシミュレータの処理要素である論理プロセス (logical process, 以下 lp) にマッピングして並列にシミュレートを行う. この方法でシミュレートすれば, 複数の lp で変数を共有する必要がないため, 分散メモリ型の並列計算機でも問題なくシミュレーションを行うことができる. ただし, シミュレート中に事象の遷移を知らせるメッセージを lp 間で送受信することになる. これらのメッセージの大きさは比較的小さいが多数送出されるため, 密結合でない汎用の並列計算機を用いると大きな性能劣化を招く可能性がある.

一方で, 実際のシミュレート対象システムは多くの pp に分割することが不可能であったり, 利用可能な lp 数が pp 数に対して不十分であるなどして, 各 lp に比較的大きな処理コストを割り当てる場合が多い. このような場合, メッセージを受信する lp は他の lp から到着するメッセージがなくてもある程度の事象処理を行うことができる. そのため, メッセージを送信する lp は受信 lp のメッセージ待ち時間を増やさない範囲でメッセージ送信を行えばよい. このメッセージ送信時期の自由度を利用すれば, 送出される小さなメッセージを取りまとめてメッセージ送受信の効率化を図ることができる. 従来の PDES に関する研究としては, 専用シミュレータの開発^{3)~6)}や, 仮想時刻同期手法の提案・評価^{1),7)~9)}などがあげられるが, 事象を含むメッセージ送受信に関する効率化の研究は行われていなかった.

本論文は, PDES のメッセージ送出力効率化の指標となる EMSR (Efficient Message Sending Rule) を提案する. EMSR は, 各 lp 間の仮想時刻同期の必要性から見て不必要なメッセージの送出を抑えるものであり, 従来の PDES におけるメッセージ送出力の効率化を目的とする. この EMSR を評価するために, 仮想時刻同期手法の中で最も基本的なヌルメッセージ法のメッセージ送出力を EMSR の見地から改良し, 待ち行列シミュレータを AP1000¹⁰⁾上に作成した. このシミュレータを用いて数百個の待ち行列で構成されるシステムのシミュレーションを行った結果, 従来のメッセージ送出力形態をとったシミュレータに比べ飛躍的な速度向上を得ることができた.

本論文は 5 章から構成される. 2 章では PDES のメッセージ送受信に関連する事項として, 基本アルゴリズムおよび仮想時刻同期手法について述べ, 3 章で効率的なメッセージ送出力 EMSR を提案する. また, EMSR に従ったメッセージ送出力の評価を 4 章で行い, 最後に今後の展望について述べる.

2. 背景

2.1 PDES の基本アルゴリズム

離散事象システムのシミュレーションでは, 事象 (システム内に生じる状態変化) を登録しておく事象リストを設け, これに登録されている事象を生起した仮想時刻 (シミュレート時刻) 順に処理していく.

事象の処理は新たな事象の発生をとまうが, 発生事象の内容はそのときのシステムの状態に依存するので, ある状態変数を参照/更新する事象は必ず生起時刻順に処理しなければならない. 逆に, 異なる状態変数を参照/更新する事象は必ずしも生起時刻順に処理する必要がない. そこで, PDES ではシミュレート対象システムを状態変数の独立した pp のネットワークで表現し, これらを各 lp にマッピングする. 各 lp は割り当てられた pp で生じる事象を生起時刻順に処理すれば, 状態変数による処理の制約を受けずにシミュレーションを行うことができる.

ただし, 事象処理過程で他の lp に属する事象を生成した lp は, 該当する lp にその事象を含むメッセージを送出する. 各 lp はこの送出されるメッセージに含まれる事象も含めて生起時刻順に事象処理を進める必要がある. そのため, メッセージを送出する lp は, 割り当てられている pp の性質や事象リストの内容などから, 今後送出する事象の生起時刻の最小値を計算し, メッセージの時刻印として付加する. これを送信 lp の受信 lp に対する保証時刻と呼ぶと, 受信 lp は保証時刻以前に生起した事象のみを処理していくことにより, その lp に属するすべての事象を生起時刻順に処理することができる.

このように, 事象を含むメッセージのみを仮想時刻同期の手段としたシミュレーションの進行は PDES の基本となるため, これを PDES の基本アルゴリズムと呼ぶことにする.

2.2 仮想時刻同期手法

PDES の基本アルゴリズムのみでは, シミュレーションを正常に終了できない場合がある.

たとえば図 1 のようなメッセージ授受関係のある 3 つの lp があり, シミュレートを実行したとする. 3 つの lp はすべて他の lp からメッセージを受信する可能性があるため, シミュレート開始直後のそれぞれの lp の被保証時刻はシミュレート開始時刻になる. よって, 他の lp からメッセージを受信しない限り事象処理を行うことができない. しかし, 事象処理を行うことができなければ他の lp に属する事象も発生しないため, メッセージ送出力が行われずデッドロックに陥る.

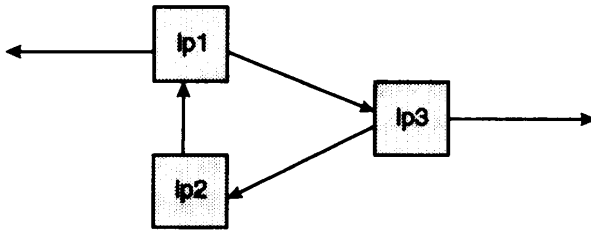


図1 デッドロックの一例
Fig. 1 A case of deadlock.

このように、PDESの基本アルゴリズムのみでシミュレートを行うと、メッセージ授受関係に有向ループを形成しているlpが存在する場合、デッドロックを生じてしまう。また、他のlpに属する事象が希にしか発生しないlpがあった場合、受信lpのメッセージ待ち時間が増えて処理効率が非常に悪くなる場合がある。

これらの問題から、事象を含むメッセージによる仮想時刻情報の伝達以外にも、何らかの形で仮想時刻同期を行う必要がある。この同期を行う手法は仮想時刻同期手法 (Virtual Time Synchronization Methods) と呼ばれ、非常に多くの研究^{1), 7)~9)}がなされてきた。

仮想時刻同期手法にはさまざまな手法が存在するが、これらの中でChandy and Misraのヌルメッセージ法¹¹⁾は最も基本的な仮想時刻同期手法である。

ヌルメッセージ法は、PDESの基本アルゴリズムで用いられるメッセージのほかに、事象を含まないヌル(空)メッセージを送出することにより、仮想時刻同期を行う手法である。基本アルゴリズムでは、他のlpに属する事象が発生したときにメッセージの送出手が行うが、ヌルメッセージ法では送出手が発生してなくても、メッセージ送出手lpに対する保証時刻が更新できる場合はヌルメッセージを送出手する。

たとえば図1の場合、各lpはメッセージ送出手lpに対する保証時刻を計算し、ヌルメッセージを送出手する。ヌルメッセージを受信したことによって各lpは被保証時刻が更新されるため、さらに送出手lpに対する保証時刻を更新することができる。このように、各lpは受信する事象がなくても仮想時刻を進めることが可能になり、その結果デッドロックも防ぐことができる。

他にもさまざまな仮想時刻同期手法が提案されているが、これらの手法については文献1)を参照されたい。

3. 効率的なメッセージ送出手則

3.1 PDESのメッセージ送出手

仮想時刻同期手法はPDESの性能に大きく影響することが知られているが、いずれの手法を用いても事象を含むメッセージの送出手は基本アルゴリズム同様、他のlpに属する事象が生じた時点でされる。

一方、離散事象システムシミュレーションでは、1つの事象処理にともなって新たに発生する事象数は少ないのが一般的である。このため、他のlpに属する事象が発生する度にメッセージ送出手を行うのでは、非常に小さなメッセージを多数送出手することになる。このことは、メッセージ送出手に対するコストが小さくない(密結合でない)並列計算機上でシミュレーションを行った場合に大きな性能劣化につながることを意味する。

この問題を解決するために、以後メッセージの送出手は送出手すべき事象が発生した時点よりも、むしろ受信lpが送出手される事象を必要とする時点までに行えばよいと仮定する。各lpに割り当てるppの処理コストが小さい場合、送出手すべき事象が発生したと同時に送出手先lpでその事象を必要とする確率は高くなる。しかし、lpに多数のppや処理コストの大きいppを割り当てる実際的な場合、1回の被保証時刻の更新によって処理可能となる事象の処理コストは大きくなる。そのため、必ずしもただちにメッセージ送出手をする必要はなくなり、メッセージ送出手のタイミングに自由度が生じてくる。この自由度を利用してメッセージ送出手の効率化を図れば、疎結合の並列計算機を用いても高速なシミュレーションが期待できる。メッセージ送出手のタイミングはメッセージ送信コストと各lpのメッセージ待ち時間に大きく影響するため、これら2つの点に分けて考察する。

3.2 lpのメッセージ送信コスト

複数の事象を同一のlpに送出手する場合、それぞれの事象を1つのメッセージとして送出手するよりも、いくつかの事象を1つのメッセージとしてまとめて送出手した方がメッセージ送信コストは小さくなる。しかし、前述のように1つの事象処理にともなって発生する事象は少数である。このため、他のlpに属する事象をまとめて送出手するには、送出手すべき事象が発生してもすぐには送出手せず、いくつか発生した時点で1つのメッセージとして送出手することになる。

3.3 lpのメッセージ待ち状態

図2のように3つのlpがあり、2つのlpが1つのlpに対してメッセージを送出手する場合を考える。この

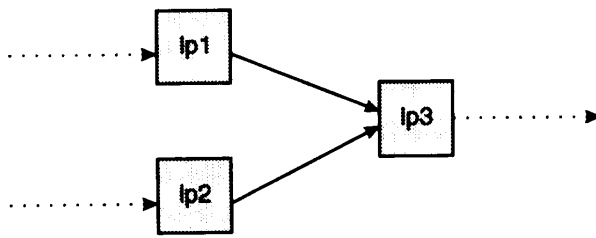


図2 3つのlpの結合関係の例

Fig. 2 An example of three lps' connectivity.

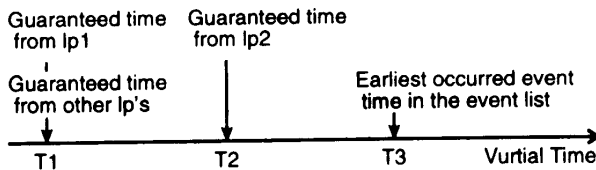


図3 図2におけるlp3の状態

Fig. 3 An lp3's status at Fig. 2.

ときのlp3の仮想時刻に関する状態の例を図3に示す。図3中、 T_1 、 T_2 はそれぞれlp1、lp2のlp3に対する保証時刻、 T_3 はlp3の事象リストに登録されている最早生起事象の時刻とする。lp3に対する接続は、lp1とlp2では本質的な違いがないため、保証時刻値の関係を $T_1 \leq T_2$ とすると、考えられるlp3内の仮想時刻の関係は次の3通りになる。

- (1) $T_3 < T_1 \leq T_2$
- (2) $T_1 \leq T_3 \leq T_2$
- (3) $T_1 \leq T_2 < T_3$

ただし、lp3の事象リストが空の場合は $T_3 = \infty$ として(3)に含めるとする。

これらすべての場合において、lp3の他のlpからの被保証時刻は T_1 となる。lpでメッセージを受信する必要があるのは、次に処理すべき事象の生起時刻が被保証時刻よりも大きい、lp内に処理すべき事象がない場合であるから、(1)の状態はメッセージを受信しなくても事象処理が進められる。残りの2つの状態は事象処理を行うことができないため、メッセージの受信を試みる必要がある。しかし、たとえば(3)の状態でlp2からのメッセージを受け取り、lp2の保証時刻 T_2 が登録事象の生起時刻 T_3 を上回っても、単に(2)の状態になるだけで結局事象処理を行うことができない。このように、いずれの場合もlp1からの保証時刻 T_1 が更新されない限り、lp3のほかlpからの被保証時刻は変化しないため、lp3の事象処理は進められないことになる。

以上から、メッセージ待ち状態は現在の被保証時刻を決定している(最も小さな保証時刻を与えている)lpからのメッセージのみによって回復する可能性があ

り、その他のlpからメッセージを受け取っても状態は変化しないといえる。

3.4 EMSR

メッセージ送信コストを考えた場合、なるべく送出事象を取りまとめてメッセージ送出を行った方が効率が良い。しかし、lpのメッセージ待ち状態を考えた場合、仮にメッセージを受信するlpが当該lpからの保証時刻を更新できないためにメッセージ待ち状態になっていたとすると、受信lpのメッセージ待ち時間を引き伸ばすことになってしまう。そこで、送出事象の取りまとめは受信lpの事象処理の進行を妨げない範囲で行うとすると、メッセージ送出は次のようになる。

「受信lpがメッセージ待ちの状態にあり、かつ自lpのメッセージ送出によって受信lpの被保証時刻が更新される場合はただちにメッセージを送出する。そうでない場合は、事象処理を進めていくうえで生ずる他の送出事象と合わせてメッセージ送出する。」

この送出則に従う限り、メッセージ送信lpは受信lpの事象処理を妨げずに送出事象の取りまとめを行えるので、メッセージ送信はきわめて効率的になる。本論文ではこれをEMSR (Efficient Message Sending Rule)と呼び、PDESの効率的なメッセージ送付則と定義する。

4. 評価

4.1 受信統計に基づくメッセージ送出

前述のように、PDESでは何らかの仮想時刻同期手法を用いなければシミュレーションを正常に終了することが保証されない。本論文では仮想時刻同期手法の中でも最も基本的なヌルメッセージ法を用い、これのメッセージ送出をEMSRに従って効率化することによってEMSRの評価を行うことにする。

EMSRによるメッセージ送出の効率化で問題となるのは受信lpの状態を知る方法である。受信lpの被保証時刻はそのlpにメッセージ送出を行うすべてのlpからの保証時刻の最小値である。このため、各送信lpは自lpのメッセージ送出によって受信lpの被保証時刻が更新されるかを判断することは(分散メモリ型計算機では)難しい。ここでは、各lpのメッセージ受信に関する統計をとり、これを基にして確率的にEMSRに近いメッセージ送出のタイミングを決定することを考える。

一般に一事象の処理コストはほぼ等しいので、一定事象数を処理する度にメッセージ送出すれば一定の実時間間隔でメッセージ送出ができる。そこで、各lpはs事象処理するごとにメッセージを送出すると仮定す

る。(ただし、事象処理ができなくなった場合は、デッドロックを防ぐため s 値にかかわらずヌルメッセージを送出するとする。) $s = 1$ のとき、メッセージ送出手法は Chandy and Misra のヌルメッセージ法のそれと同一になり、 s の値が大きくなるに従って事象の取りまとめが多く行われるようになる。よって、なるべく s の値を大きくした方がメッセージ送出手法は良くなる。ただし、EMSR では自 lp のメッセージ送出手法によって受信 lp のメッセージ待ち状態を回復できるような場合、ただちにメッセージ送出手法をしなければならない。 s の値が必要以上に大きいと、受信 lp のメッセージ待ち状態を長引かせる確率が高くなる。

そこで、受信 lp は i 番目の lp から送出手法されたメッセージの受信数 r_i と、さらに、メッセージ待ち状態で i 番目の lp から送出手法された被保証時刻を更新するメッセージの受信数 r_{wi} を記録しておく。 r_{wi}/r_i が大きい、すなわち i 番目の lp が当該 lp をメッセージ待ちにしてしまう確率が高い場合、その lp に対してメッセージ送出手法の増加要求メッセージを送出手法する。送信 lp はなるべく s の値を大きくしておき、メッセージ送出手法の増加要求メッセージを受信する度に s の値を小さくしていく。送信 lp は、受信 lp から増加要求を受け取らなくなった時点で、確率的に EMSR に従ったメッセージ送出手法を行う s 値を得たことになる。

これをシミュレーション初期の段階で行い、すべてのメッセージ送信 lp の設定している s 値が受信 lp の r_{wi}/r_i 値をしきい値かそれ以下にしている場合、統計収集を終了する。その時点で設定されている s の値に従って残りのシミュレーションを行えば、EMSR に近いタイミングでメッセージ送出手法を行うことができる。本論文では、この事象間隔 s を用いたメッセージ送出手法を受信統計に基づくメッセージ送出手法と呼ぶことにする。

4.2 他のメッセージ送出手法との比較

ヌルメッセージ法は、送信 lp が受信 lp の状態にかかわらず (ヌル) メッセージ送出手法を行う能動的なメッセージ送出手法である。これは、受信統計に基づくメッセージ送出手法において、つねに $s = 1$ であることに相当する。この能動的なメッセージ送出手法は、メッセージ受信 lp の必要としない (ヌル) メッセージを多数送出手法してしまう可能性がある。この問題を回避するために、送信 lp は通常ヌルメッセージを送出手法せず、受信 lp から保証時刻の問合せメッセージを受信したときに限りヌルメッセージを送出手法する問合せメッセージ法^{7),8)}が提案されている。

問合せメッセージ法は、自 lp 内の事象処理ができな

くなった場合、被保証時刻を決定している lp に対して問合せのメッセージを送出手法する。これを受信した lp は、問合せメッセージを送出手法してきた lp への保証時刻が更新できる場合、ヌルメッセージを送出手法する。その時点で保証時刻が更新できない場合は事象処理を進め、保証時刻が更新できるようになってから (ヌル) メッセージを送出手法することになる。ただし、(ヌル) メッセージを送出手法する前に事象処理ができなくなった場合は、さらにその lp の被保証時刻を決定している lp に対して問合せメッセージを送出手法する。

問合せメッセージ法では、ヌルメッセージ送出手法は受信 lp からの要求駆動であるため、受動的なメッセージ送出手法といえる。ただし、事象を含むメッセージの送出手法は基本アルゴリズムと同様であるから能動的なメッセージ送出手法である。

事象を含むメッセージ送出手法も要求駆動とする、すなわちすべてのメッセージを受動的とした場合、問合せメッセージ数は増加するが事象を含むメッセージの取りまとめが可能となる。以後、すべてのメッセージ送出手法を受動的としたものを単に受動的メッセージ送出手法と呼ぶことにする。本論文で前提としている各 lp に割り当てる処理コストが比較的大きい場合、メッセージの取りまとめは重要であり、受動的なメッセージ送出手法の方が問合せメッセージ法のメッセージ送出手法より有利と考えられる。後述する評価モデルをシミュレートした結果では、16 個以上の lp を用いた場合に問合せメッセージ法の方がメッセージ送出手法数が少なかったが、実行時間は lp 利用数にかかわらず、つねに受動的なメッセージ送出手法の方が 5% 程度高速であった。

また、受動的なメッセージ送出手法は能動的なメッセージ送出手法と送出手法が対照的であり、メッセージ送出手法の比較として問合せメッセージ法のそれよりも考察が行いやすいと思われる。以上から、本論文ではこの受動的なメッセージ送出手法、能動的なメッセージ送出手法と受信統計に基づくメッセージ送出手法を比較して EMSR の評価を行うことにする。

4.3 待ち行列シミュレータの実装

本論文では、メッセージ送出手法形態を評価するにあたり、離散事象システムのモデル化において基本となる待ち行列システムをシミュレート対象とした。待ち行列システムは、一般にネットワークを構成する待ち行列がそれぞれ独立した状態変数 (サーバの空き状況、待ち行列の長さなど) を持つため、多くの pp で表現することが容易である。さらに、一事象の処理コストが小さく、適用した仮想時刻同期手法の影響が出やすいことから、これらの手法の評価によく用いられてい

る。そこで、能動的なメッセージ送出、受動的なメッセージ送出、受信統計に基づくメッセージ送出による待ち行列シミュレータを高並列計算機 AP1000¹⁰⁾上に実現した。

受信統計に基づくメッセージ送出ではいくつかのパラメータが必要となる。AP1000 は比較的高速なネットワークを持っているため、実際の s 値が EMSR に従った s 値に比べて多少小さくなるより、むしろ受信統計処理の長引く方が実行時間に与える影響は大きいと考えられる。そこで、メッセージ送出頻度の増加要求を出す r_{wi}/r_i のしきい値を 0.1 に設定し、さらに、1 回のメッセージ送出頻度の増加要求に対し s の値を半分にして、早めに s 値が収束するように設定した。

また、統計はシミュレート終了仮想時刻の 1/1000 ごとに収集して、 s が収束するまでのシミュレートも含めて実行時間を計測した。さらに、本来 EMSR では各 lp に割り当てられている pp の処理コストが比較的小さい場合を考えていない。このため、EMSR に従った効率化を目的とする受信統計に基づくメッセージ送出では、割り当てられた pp の処理コストが小さいとメッセージ受信統計が不安定になり、不要なメッセージ送出頻度の増加要求を出してしまう。そこで、本論文ではメッセージ送出頻度増加要求を受けてもある値より s が小さくならないための下限値を設定した。 s の下限値はモデルの性質、シミュレータの事象処理コストとメッセージ送信コストの比などに依存すると考えられるが、実際にトポロジーの異なるいくつかの待ち行列モデルをシミュレートした結果、モデルによる最適値のずれは少なく、すべての場合で 6 から 8 程度が最適であった。そこで、今回の評価では $s = 8$ を下限値としてすべてのシミュレーションを行った。

4.4 評価モデル

本論文では、 16×16 の二次元メッシュ状に結合した閉鎖型待ち行列ネットワークモデル (図 4) を評価に用いた。各待ち行列は FCFS (First Come First

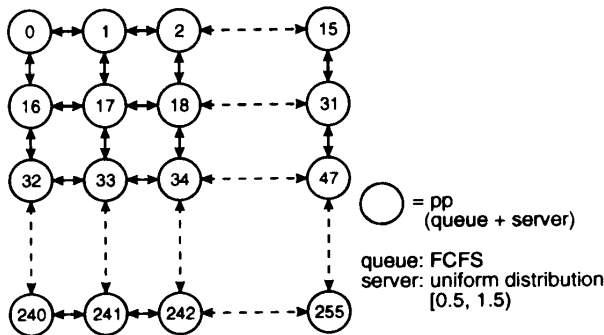


図 4 評価用待ち行列ネットワークモデル
Fig. 4 A queuing network model for evaluation.

Served) とし、1 つのサーバから近隣のサーバへの移動確率はそれぞれ均等に設定した。また、各サーバのサービス時間は $[0.5, 1.5]$ [ut] の均一分布とした。ただし、ut は仮想時刻の単位時間とする。このようなモデルを評価に用いたのは、各待ち行列がほぼ平等に接続されているため、PDES のもうひとつの課題である pp のネットワーク分割および lp へのマッピング^{12),13)}が、負荷分散や lp 間を遷移する事象数の最小化などを考えても容易に行えるためである。

4.5 評価結果

AP1000 は、従来の DMA 転送を用いるメッセージ送受信機構 (Normal mode) のほかに、ラインセンスド/バッファレシーブを用いて高速な通信を行う機構 (LS+BR mode) を持つ。本論文では、両方の通信機構を用いてシミュレーションを行い実行時間を測定した。評価モデルを 10,000 [ut] シミュレートした際の速度向上率を図 5、図 6 に示す。ただし、速度向上率の基準となる 1 個の lp での実行時間には、メッセージ

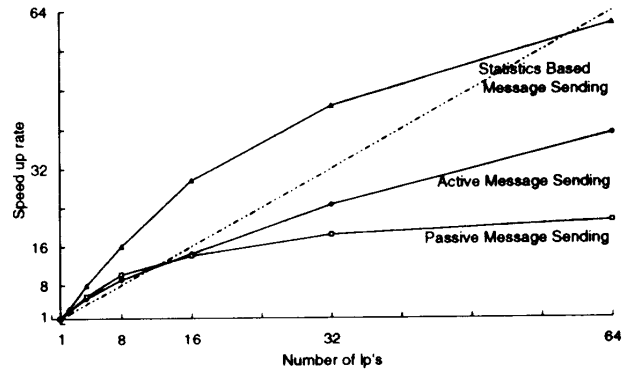


図 5 LS+BR mode での速度向上率
Fig. 5 Speed up rate in LS+BR mode.

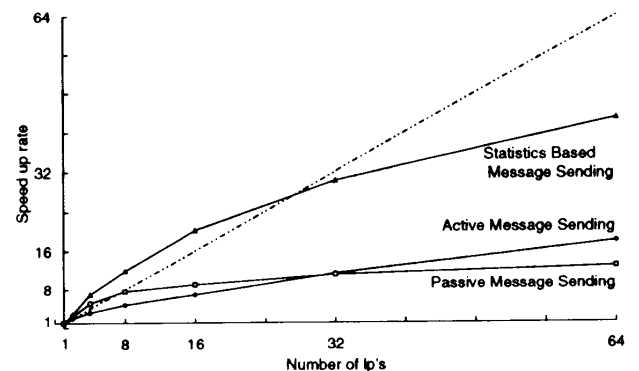


図 6 Normal mode での速度向上率
Fig. 6 Speed up rate in Normal mode.

送出や保証時刻計算など PDES 特有の処理を省いた逐次処理用のシミュレータによる実行時間を用いた。

図5, 図6中, 利用lp数以上の速度向上率を得ている部分がある。ここで, その原因を調べるために各lpの事象リストについて考えてみる。今回の評価モデルでは各サーバのサービス時間最大値が1.5[ut]であるので, 事象リストに登録されている事象の生起時刻分布が1.5[ut]幅の均一分布であると仮定する^{*}。各サーバのサービス時間平均値は1.0[ut]であるから, 上記の仮定により新たな事象の事象リストへの挿入位置は平均的に先頭から約2/3の位置になる。本論文で実装したシミュレータの事象リストは一般的な単方向連結リスト構造であるので, 一事象に登録するために他の事象の生起時刻と比較する回数は, すでにリストに登録されている事象数の約2/3となる。つまり, 事象登録コストは事象リストに登録されている事象数と比例関係にあるといえる。事象リストに登録されている事象数は, マッピングされているモデルの大きさにほぼ比例するので, n 個のlpを用いた場合には約 $1/n$ になる。結局, 事象リスト登録コストは約 $1/n$ となり, さらに各lpで処理する必要のある事象数が約 $1/n$ になるため, 利用lp数以上の速度向上率を得ているものと思われる。事象登録コストのlp数による変化を抑えるには, 事象リストにヒープ構造を用いるなどの必要がある。

図5, 図6の線の傾きから, 受動的なメッセージ送出が利用lp数の増加にともない, 能動的なメッセージ送出や受信統計に基づくメッセージ送出よりも速度向上が図れていないことが分かる。これは, 利用lp数が増加すると, 保証時刻の問合せを行ったlpの返答を受信するまでに要する時間が長くなり, 全体のメッセージ待ち時間が増加して引き起こされたものと思われる。これを検証するために, 32個のlpを用いて500[ut]のシミュレーションを行い, 各lpの処理を分析した結果を図7に示した。ただし, 受信統計に基づくメッセージ送出では統計収集間隔を他の実行と同一の $10,000/1000 = 10$ [ut]にした。図7から, 受動的なメッセージ送出は他の2つのメッセージ送出よりもメッセージ受信およびメッセージ待ちの時間が長いことが分かる。また, メッセージ送受信ライブラリのコストを除いても受信統計に基づくメッセージ送出が能動的メッセージ送出の約半分で行われているのは, (ヌル)メッセージ送信ごとに行われる保証時刻計算

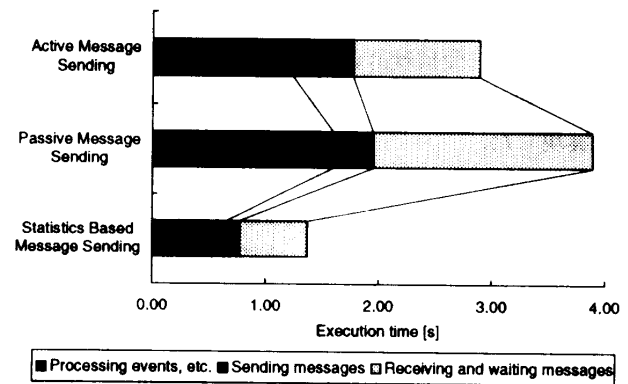


図7 実行時間の分析結果 (32lp's)
Fig. 7 Breakdowns of execution time (32lp's).

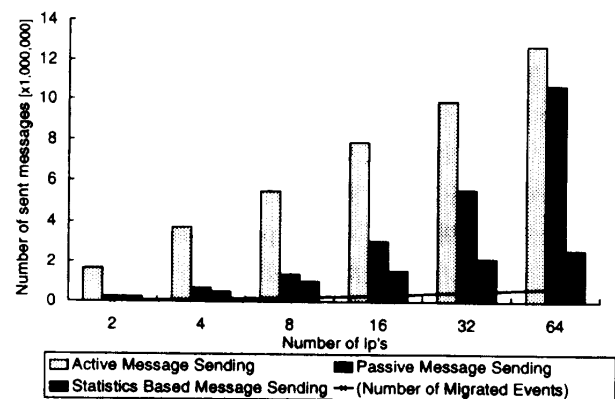


図8 LS+BR modeでのメッセージ送出数
Fig. 8 Number of sent messages in LS+BR mode.

の回数が増加しているためと考えられる。

一方, 図5と図6を比較すると, 能動的なメッセージ送出はNormal modeに変更した場合に性能劣化が大きいといえる。8個のlp使用時のNormal modeでの実行時間は, LS+BR modeの実行時間を基準とした場合, 能動的なメッセージ送出が191.6%, 受動的なメッセージ送出が134.2%, 受信統計に基づくメッセージ送出が134.9%となった。これは, 能動的なメッセージ送出が他のメッセージ送出に比べて, 送出メッセージ数が多いためと思われる。LS+BR modeでシミュレートした際に発生したメッセージ数を図8に示した。ただし, 受動的なメッセージ送出では問合せメッセージ数も加えているため, 実際に行われた仮想時刻同期は約半分である。また, 図8中, 実線は実際にlp間を遷移した事象数を示す。図8から, 能動的なメッセージ送出のメッセージ送出数が非常に多く, これが性能劣化に大きな影響を与えていることが分かる。

図8ではLS+BR modeでの発生メッセージ数を示したが, Normal modeでシミュレートした際に発生

^{*} 実際には, 他のlpから受信した事象が原因となって, 事象リストに登録されている事象の生起時刻分布は1.5[ut]幅以上あり, かつ後方に膨れた分布になると考えられる。

したメッセージ数も LS+BR mode のそれとほとんど変わらなかった。受信統計に基づくメッセージ送出の s 値は基本的に事象処理コストとメッセージ送出コストの比によって変化するはずであるが、今回の比較では後述する s 値の変動の方が大きく、送出機構の変化による大きな s 値の変動は見られなかった。

以上の結果をふまえると、受信統計に基づくメッセージ送出はメッセージ待ち時間が少なく、またメッセージ送出数も少ないため、メッセージ送信コストが大きい場合にも性能劣化が小さいといえる。

受信統計に基づく s 値の収束は比較的早く、収束が最も遅かった 2 個の lp を用いた場合でも 5~6 回で s 値が決定した。 r_{wi}/r_i の値はシミュレーション開始時点ではどの場合も 0.8 以上であり、収束直前までは 0.6 以上を保つが、その後一回の s 値の半減によって 0.1 程度になった。結局、今回の評価モデルでは r_{wi}/r_i のしきい値を 0.1 としたが、このしきい値を 0.3 程度にしても同じ結果が得られたといえる。収束後の s 値は 2 個の lp 利用時で 32~24、4 または 8 個の lp 利用時で 23~16 となった。 s の値は実行のタイミングによって変化したが、これは s 値の収束を早めるために s 値を半減させていったことによるものと思われる。ただし、 s 値変動による実行時間の変化はどの場合においても 1% 以内であった。

設定した下限値 ($s = 8$) は 32 個の lp を用いた時点で実行時間に影響が出はじめた。このとき、約 30% の出力リンクの s 値が下限値に設定された一方で、いくつかの出力リンクの s 値は下限値の 2 倍以上に設定されていた。評価に用いたメッシュ状のモデルは外枠にあたる部分以外はすべて平等に結合されており、送出事象の発生間隔もすべての入力リンクで同一であるため、本来すべての出力リンクの s 値はほぼ同一に設定されるはずである。この結果は 4.3 で指摘した、各 lp に小さな処理コストが割り当てられた際に生ずる受信統計の不安定さを示すものである。

一方、64 個の lp を用いた場合、すべての出力リンクの s 値は 8 (下限値) に設定されており、下限値を設定しない場合は $s = 1$ となって能動的なメッセージ送出と同一になった。つまり、64 個の lp を利用した場合では、各 lp はつねに他の lp からのメッセージ待ち状態にあり、受信統計を見る限りはメッセージの取りまとめを行うべきでないといえる。この場合の下限値は、メッセージハンドリングのオーバーヘッドを減少させるという別の役目を果たし、結果的に大幅な速度向上率を得ていたといえる。

マルチメッセージ法 (能動的なメッセージ送出) では、

他の lp に対する保証時刻が更新できる場合マルチメッセージを送出する。lp 内に処理可能な事象がない場合は、受信統計に基づくメッセージ送出も s 値にかかわらず同様にマルチメッセージを送出するので、今回の評価モデルで利用 lp 数を pp 数と同数にするなど、連続的に事象処理が行えない状況を作れば、受信統計に基づくメッセージ送出は下限値にかかわらず能動的なメッセージ送出と同一のメッセージ送出形態になるはずである。しかし、上記の結果から、受信統計に基づくメッセージ送出は比較的 lp 数が少ない場合に非常に効率的であることが分かった。これは、実際的な PDES において EMSR がメッセージ送出の効率化を行う際の正しい指標であることを示している。

EMSR に従う際の問題点としては、各 lp の負荷が極端に異なる場合、負荷の高い lp が負荷の低い lp に対してメッセージ送出を頻繁に行い、効率の悪いメッセージ送出になってしまうことである。特に、受信 lp の負荷が低く、さらにその lp の事象処理の滞りが他の lp の事象処理に影響しない (メッセージ送出を行わない) 場合、EMSR ではメッセージの効率化を達成できない。これは EMSR の問題点といえるが、このような状況は pp の分割および lp へのマッピングに失敗した際に生じることであり、モデルの分割・マッピングが PDES の性能に大きな影響を与えることを示している。今回の評価に用いたようなトポロジー的に分割が容易なモデルでなく、一般のシミュレート対象モデルを考えた場合には、つねに lp へのマッピングの影響を考慮する必要があるといえる。

5. む す び

本論文では、PDES のメッセージ送受信効率化の指標となる EMSR を提案した。また、これに従ってマルチメッセージ法のメッセージ送出を改良して大きな処理効率向上を得た。今回の評価では仮想時刻同期手法としてマルチメッセージ法を用いたが、EMSR は仮想時刻同期のタイミングからメッセージ送出の効率化を図るものであり、他の仮想時刻同期手法でのメッセージ送出効率化にも適用できることが予想される。たとえば楽観的な方針を持つ仮想時刻同期手法¹⁾では、各 lp はメッセージ待ちを行わないが、ロールバックは本来仮想時刻同期を行うべきときに同期を行わないために発生するものであるから、ロールバックを増やさない範囲でメッセージの取りまとめを行えば、手法に依存するロールバック回数を変化させずに効率的なメッセージ送出を行えることが期待できる。今後は他の仮想時刻同期手法を EMSR に従って効率化することに

より、送出則の一般的な有効性を示し、また、さまざまなシミュレーション対象システムおよびそれらのlpへのマッピングに対する適応性を評価していきたい。

謝辞 本研究で AP1000 の利用環境および貴重な情報を提供して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大変感謝いたします。

参考文献

- 1) Fujimoto, R.M.: Parallel Discrete Event Simulation, *Communication of the 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*, SCS, pp.106-114 (1993).
- 3) 稲守, 戸田: 複数マイクロプロセッサを用いた並列形通信網トラヒックシミュレータの評価, 電子通信学会論文誌, Vol.J68-B, No.1, pp.22-29 (1985).
- 4) 佐竹, 上月, 西田, 宮原, 高島: 分散型待行列網シミュレータ, 電子通信学会技術研究報告 EC83-40, 電子通信学会, pp.25-34 (1983).
- 5) 佐藤, 中西, 真田, 手塚: 並行処理型待行列網シミュレータ D-SSQ, 電子通信学会論文誌, Vol.J69-D, No.3, pp.302-311 (1986).
- 6) 中川, 小林, 相磯: データ駆動型離散系シミュレータ KDSS-I, 電子通信学会論文誌, Vol.J65-D, No.3, pp.386-393 (1982).
- 7) Misra, J.: Distributed Discrete-event Simulation, *ACM Computing Surveys*, Vol.18, No.1, pp.39-65 (1986).
- 8) 工藤, 木村, 天野, 寺沢: 問合せに基づく並列論理シミュレーションアルゴリズム, 電子情報通信学会論文誌, Vol.J75-D-I, No.4, pp.221-231 (1992).
- 9) 松本, 瀧: バーチャルタイムによる並列論理シミュレーション, 情報処理学会論文誌, Vol.33, No.3, pp.387-395 (1992).
- 10) 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).
- 11) Chandu, K.M. and Misra, J.: Distributed Simulation: A case study in design and verification of distributed programs, *IEEE Trans. Software*

Engineering, Vol.5, No.5, pp.440-452 (1979).

- 12) Nandy, B. and Loucks, W.M.: An Algorithm for Partitioning and Mapping Conservative Parallel Simulation onto Multicomputers, *6th Workshop on Parallel and Distributed Simulation*, SCS, pp.139-146 (1992).
- 13) 高井, 根本, 成田: 待ち行列並列シミュレーションのマッピング手法, 情報処理学会研究報告 ARC105-1, HPC50-1, 情報処理学会, pp.1-8 (1994).

(平成7年6月29日受付)

(平成7年12月8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。平成6年同大学大学院理工学研究科電気工学専攻修士課程修了。現在同大学大学院博士課程在学中。平成7年情報科学研究教育センター助手。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。現在同大学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。37年アメリカ・パデュー大学大学院留学(フルブライト留学生)。38年早稲田大学理工学部助手, 以後, 講師・助教授を経て, 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。