タイムワープ法を用いた離散事象並列シミュレータにおける仮想時刻の同調

4L－2

根本貴由　高井峰生　黄強　成田誠之助
早稲田大学理工学部

1. はじめに

情報ネットワーク網や道路交通網、工場の生産ラインなど、離散事象システムとして表現できるものも多々存在する。これらのシステムの振舞いを調査する手段として、離散事象シミュレーションは重要である。しかし、近年インターネットの爆発的流行など、これらのシステムは大規模化の一途を辿っている。この様なシステムをリアルタイムにシミュレートする要求なども増しているため、より高速なシミュレータの登場が切望されている。そこで、離散事象システムの空間的局所性を用いて、シミュレートを並列化し、高速化する研究[1) 2) 3)]が多くのところでなされてきた。

2. タイムワープ法

シミュレータの並列化にともなって必要となるのが、並列計算機の処理要素(PE：Processing Element)間で各サブモデルのシミュレーション同士の正当性を保証する処理である。これを仮想時刻同期処理といいこれまでにいくつかの手法が提案[2) 3)]されている。これらの手法は、保守的手法と楽観的手法とに大別される。離散事象シミュレーションにおいては事象処理を繰り返すことで処理を進行させるが、前者では、正当性を保証された事象処理のみを行い、シミュレートを進行させる。一方、タイムワープ法が分類される後者では、事象の正当性が保証されない場合でも事象処理を行い、矛盾が発生した時点で以前の状態に戻すこと(Roll Back)で、シミュレーション全体としての正当性を保つ。

```
while (GVT < Endtime) {
  Process Event;
  Handle Message;
  if (GVT is to be updated) {
    barrier synchronization;
    GVT = Min(LVTs of All PEs);
  }
}
```

図1 タイムワープ法の処理手順

タイムワープ法の処理手順は図2で表わされる。図中、LVTは各PEでの局所的仮想時刻(Local Virtual Time)、GVTは全PEにおけるLVTをまとめた広域仮想時刻(Gloval Virtual Time)である。

3. 仮想時刻同調の実現方法

シミュレートを並列に行う場合、それぞれのPEが異なるLVTをもつ。並列化に伴うオーバーヘッドであるRoll Back処理の発生は、あるPEで処理が遅れていることが大きな要因となる。この処理の遅れは、LVTの遅れとして現れるので、LVTをなるべく同じような値になるように同調させることで処理効率の向上が期待できる。具体的には、次回のGVT更新時までに処理する事象数x'を

式1：`x' = x - x * (LVT - GVT) / diffLVT`

で決定する。ここで`diffLVT`は前同期間隔でのLVT進行幅である。

4. 評価条件

本方式を評価するにあたり、比較対象のシミュレータとしてはもっとも基本的なタイムワープ法を用いたシミュレータを選んだ。この基本的なシミュレータに仮想時刻の同調をさせる機能を付加し、それぞれで実際にシミュレートさせ、評価を行った。

シミュレート対象モデルとしては、分割&マッピング(以下、マッピング)の影響を回避するため、単純な構造であり、最適なマッピングが容易なトーラス型のモデルを採用した。サービスの型が異なるモデル(1024(＝32×32)ノード)を13種類用意し、PE数(1、2、4、8、16、32、64)によって格子状にマッピングする。シミュレータを導入したのがPE間トーラス結合である富士通研究所のAP1000であるので、PE間通信の点から考慮してもこのマッピングが最適であるといえる。

Controlling Virtual Times for Time Warp on Parallel Discrete Event Simulator
Takayoshi Nemoto, Mineo Takai, Huang Qiang, Seinosuke Narita
School of Science and Engineering, WASEDA University
3-4-1 Ohkubo, Shinjuku, Tokyo 169 department of electrical engineering Narita laboratory

5. シミュレート結果

シミュレート結果として各PE数での速度向上率をグラフにしたものを図2に示す。

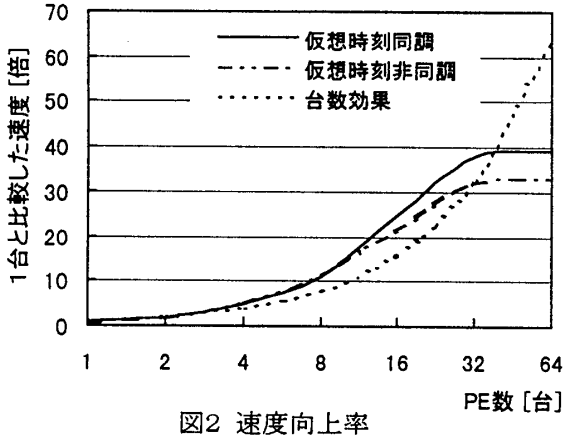

図2 速度向上率

同期待ち状態の割合を図3に示す。

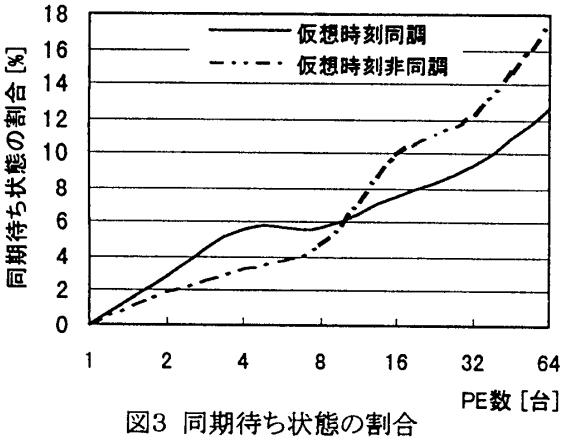

図3 同期待ち状態の割合

6. 考察

図2より、台数以上の並列効果が出ているが、これは対象モデルを分割することで事象リストが小さくなり事象を事象リストに挿入するためのコストが大幅に減少したことが最も大きな原因と考えられる。

また、同図においてPE数が少ない場合、仮想時刻を同調させた方が処理効率が低下した事が分かる。この理由はPE数が少ない程、各PEにおいて、同期間隔にある一定回数の事象処理によって進行した仮想時刻の差 diffLVT が小さくなり、式1によって更新される次の同期間隔での事象処理数x'が大幅に小さくなってしまうからであろう。これにより同期回数が増加したこと、x'の変動が激しく図3に現れているように同期待ち状態の割合が大きくなったことなどが原因と考えられる。

PE 数が増えた場合、仮想時刻を同調させた方が処理効率が約 20%程度向上している。同一シミュレート時間での Roll Back 処理回数はどちらのシミュレータでも同程度であったので、仮想時刻を同調させた方が Roll Back 処理によって過去の状態に戻される事象の数が減り無駄な処理が減少したためと思われる。また図3にもあらわれているように、仮想時刻を同調させるために事象処理数を調節した結果、事象処理コストと Roll Back 処理コストとがバランスし、同期待ちの割合も同様に 20%程減少させることができた。

7. おわりに

本稿では、もっとも基本的なタイムワープ法とそれに仮想時刻の同調をさせる機能を組み込んだものとでシミュレート結果を比較することにより評価を行った。しかし、これまでにもタイムワープ法に対する様々な工夫が研究されており [1]、それらと本稿の方法とを複合的に組み合わせた場合、より効率の向上が期待できる。今後は、様々な工夫と組み合わせた場合の評価を行っていく予定である。

謝辞

最後に、この研究を行うにあたって、素晴らしい研究開発環境を提供していただきました富士通研究所、ならびに早稲田大学村岡研究室の皆様方に大変感謝いたします。

参考文献

1) Richard M.Fujimoto, "Parallel Discrete Event Simulation", Communication of the ACM, Vol.33, No.10, pp.30-53, October 1990.

2) David R.Jefferson, "Virtual Time", ACM Transactions on Programming Languages and Systems, Vol.7, No.3, pp.404-425, July 1985.

3) Jayadev Misra, "Distributed Discrete-Event Simulation", Computing Surveys, Vol.18, No.1, pp.39-65, March 1986.