

## 待ち行列並列シミュレーションの一マッピング手法

4L-3

黄強 高井峰生 根本貴由 成田誠之助

早稲田大学理工学部電気工学科

## 1. はじめに

工場の生産ライン、交通、通信網、計算機システムなどの様な離散事象システムは、並列性を有するシステムである。近年、これらのシステムは大規模化、複雑化の傾向にある。このような離散事象システムのシミュレーション(Parallel distributed Discrete Event)の実行時間を短縮するために、並列処理技術は大変有効である\*。離散事象シミュレーションの並列処理の実行においては、対象モデルの分割・マッピングと並列計算機を構成する各プロセッサのシミュレート時刻の同期は大きな問題である。前者の並列シミュレーションの前処理としての対象モデルのマッピング問題は、シミュレーションの並列処理効率に大きな影響を与える。本稿では、前者のマッピング問題を取り上げ、モデルの空間的な位置情報を利用したマッピング手法を提案・評価する。

## 2. 提案するマッピング手法

## 2.1. PDESの並列処理

並列シミュレーションの実行に当たって、処理効率を上げるため

- 各プロセッサにおける負荷の均等化
- プロセッサ間の通信コストを最小化

が必要となる。これを式で表すと

$$|W(p) - W(q)| < \delta$$

$$\min \sum_k C(p, q)$$

ただし、 $W(p), W(q)$ は異なるプロセッサ  $p, q$ における事象処理コスト、 $C(p, q)$ はプロセッサ  $p, q$ 間の通信コスト、 $\delta$ は無限小正数とする。

## 2.2 離散事象並列シミュレーションのマッピング

待ち行列(離散事象)並列シミュレーションにおいて、シミュレーション実行における総事象処理コストは一定であるので、これを均等に配分すれば、並列処理効率はプロセッサ間の通信コストに依存する。プロセッサ間の通信コストは、分割されたサブモデルの並列性によって決まる。対象モデルのサブモデルの並列性を保てば、プロセッサ間の事象を含むデータ転送コストを減らすことができる。同時に、仮想時刻同期間

One Mapping Method for Parallel Queuing Network Event Simulation

Huang Qiang / Takai Mineo / Nemoto takayosi / Narita Seinosuke

Department of Electrical Engineering,

School of Science and Engineering, WASEDA University

隔を長くし、同期回数を減らすことで、同期のためのデータの通信コストの変動を抑えることもできる。サブモデルの並列性は、対象モデルの構成要素間の入出力関係に依存する。それをいかに引き出せるかはマッピングに大きく依存する。ゆえに、プロセッサ間の通信コストが、マッピングによって変動する。

また、分割したサブモデルをシミュレートするのに、必要な事象処理コストを均等にすることと、通信コストを最小化することを同時に行うのは非常に困難である。メッセージ・パッシング型並列計算機、一般に、一つの事象処理にかかる処理コストより、一つのメッセージを転送にかかる通信コストの方が大きい。そこで、通信コスト削減することに重点をおいた方が、並列処理効率が向上すると考えられる。この点もマッピングの際には、考慮に入れなければならない。

## 2.3 提案するマッピングアルゴリズム

本稿では、2.1.2で述べたことを考慮したマッピング手法を提案する。

待ち行列モデルは、一般に、グラフを用いて表示できることから、対象モデルの構成要素間の入出力関係を各要素のグラフにおける空間的位置情報で表すことができる。すべての要素の集合を  $V$ 、すべて要素間のリンクの集合を  $E$  とすると、モデルを  $G = (V, E)$  で表すことができる。

本アルゴリズムは、まず、モデルの構成要素間の入出力関係に注目を置き、[5]のグラフ・レイアウトアルゴリズム

$$\min \sum_{(v,w) \in E} \omega(v,w)(\lambda(w) - \lambda(v))$$

$$\text{subject to: } \lambda(w) - \lambda(v) \geq \delta, (v,w) \in E$$

を用い、構成要素を入出力関係に基づいてソートし<sup>①</sup>、構成要素入出力関係を反映する位置情報  $\text{grank}()$  を求める。ここで、 $\lambda(w) - \lambda(v)$  は、リンク  $e = (v, w)$  の長さ、 $\omega(v, w)$  はリンク  $e$  の重みとする。

次に、求めた位置情報  $\text{grank}$  に基づき、あらかじめ求めたモデルの各要素で予想される事象処理コストと入出力のための通信コストを参考に、次のようにモデル分割・配分を行う。

```
ecost = total process cost/number of sub model
For every sub model {
  current_cost = 0;
```

<sup>①</sup> 詳しい説明は[5]を参照されたい。

```

For every grank {
  Take a unselected node to the sub model;
  Increase current_cost;
}
While(current cost < ecost) {
  For every node of this sub model {
    Search for every output link of this node;
    If length of link is short {
      Take this link into the same sub model;
      Increase current_cost;
    }
    Else if (size of the output link is big
      & current_cost < ecost/2){
      Take this link into the same sub model;
      Increase current_cost;
    }
  }
}
}
}

```

ただし、ecost はシミュレーション実行で予測される事象処理コストを利用PE数で割ったものである

### 3. アルゴリズムの実装と評価

提案したマッピングアルゴリズムを実装したマッピングの評価は富士通のAP1000上に構築したシミュレータを用いて行った。サイズの異なるモデルのマッピングの効果と、同じマッピングを施した同じモデルの異なる同期手法を用いたシミュレータでのマッピング効果との二項目について、モデルを単に格子状に分割したものの実行結果と比較した。

#### 3.1. モデルサイズ別のマッピング効果

一般的に、待ち行列のサイズ(ノード数)が小さいほど、並列性が小さいことが予想される。図1から並列性の小さいモデルにも効果あることが判る。

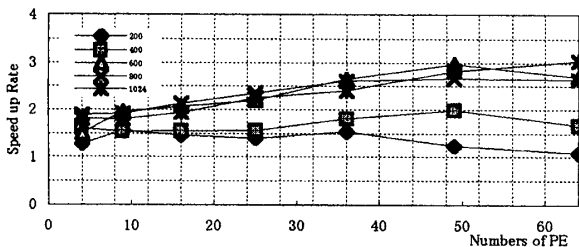


図1(a) Mapped

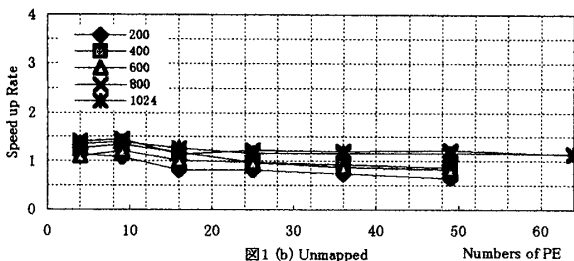


図1(b) Unmapped

図1. サイズ別のマッピング効果の比較

#### 3.2. 異なる同期手法を用いたシミュレータでのマッピング効果

図2から、ヌルメッセージ法(NULL)、問い合わせメッセージ法(STAT)、受信統計に基づくヌルメッセージ法(DMND)と同期命令法(SYNC)を用いたシミュレータで、いずれも、並列効果があることが判った。

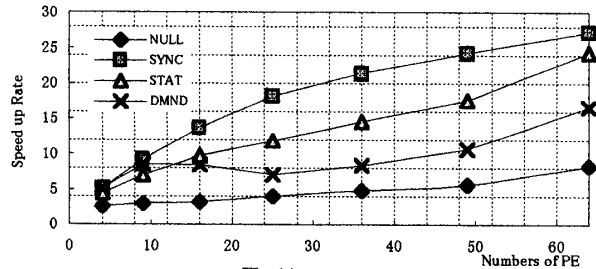


図2(a) Mapped

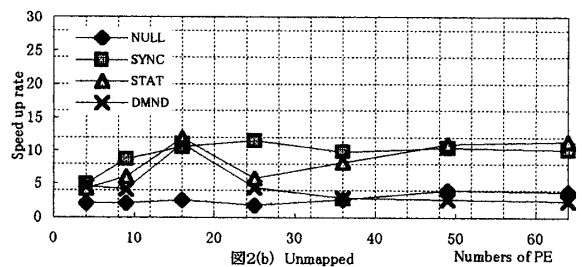


図2(b) Unmapped

図2. 同期手法の異なるシミュレータで実行結果

### 4. むすび

本稿では、待ち行列の持つ並列性を考慮し、モデル構成要素の位置情報を用いたマッピングアルゴリズムを提案した。実装し、評価を行った。その結果から、提案したアルゴリズムは待ち行列並列シミュレーションによりマッピングを提供できることが分かった。

これまで、提案するアルゴリズムの長所を述べてきた。モデルによって、マッピングの時間がかかるのが、本アルゴリズムの短所である。また、これから課題である。

[参考文献].

1. \* Fujimoto,R.M.;Parallel and Distributed Discrete Event Simulation:Algorithmsand Applications, 1993 Winter Simulation Conference,SCS,p106-114.
2. Emden R.Ganser, Eleftherios Koutsofios, ...:A Technique for Drawing Directed Directed Graphs,IEEE Transactions on software engineering ,Vol.19,No.3.p214-230
3. Fujimoto,R.M.:Parallel Discrete Event simulation , Communication of the ACM,Vol.33, No.10.p30-53.
4. Nandy,B.and Loucks,W.M.:An Algorithm for Partition ing and Mapping Conservative Parallel simulation onto Multicomputer. 6<sup>th</sup> Workshop on Parallel and Distributed simulation,SCS,p133-146