

# 交通流シミュレーションの並列処理とその評価

2T-1

塚本 健一 松岡 康広 伊藤 英則 世木 博久  
名古屋工業大学

## 1 はじめに

本稿は並列計算機上の交通流シミュレーションシステムの構成方法とこれを用いた実験と評価結果について述べる [1]。

本システムでは、道路や交差点をオブジェクト構成要素と見なし、交通網のデータベースをオブジェクト指向に基づいて構築した。このシステムを並列計算機 (AP1000) 上にインプリメントし、実際にいくつかの交通網データベースを与えてシミュレーション実験を行ない、並列効果と交通流解析ツールとしての有効性を確認した。

## 2 システムの概要

### 2.1 交通網データベース

データベースはオブジェクト指向に基づいて構成しており、交通網の構成要素を次の6つのクラスによって表現する。

Map	シミュレーションを行なう領域
Intersection	交差点
Path	車線を結ぶ交差点内の通過路
Road	交差点を結ぶ道路
Course	進行方向の同じ車線の集まり
Lane	交差点へつながる車線

オブジェクトはこれらのクラスのインスタンスであり、クラスに応じたデータメンバ (例えば、Road であれば道路の幅や長さなど) を持っている。また、クラス間には階層的な関係があるため、交通網は、オブジェクトをノードとする図1に示されるような木構造として表現される。

### 2.2 シミュレーションモデル

本システムで用いている交通流モデルでは、車線上にいる車を Lane クラスが持つキューとして取り扱っている。図2において、中の四角は車であり、数字は交差点に進入するまでに何秒かかるかを示している。左上の図において、3つの車線のキューは、上の車線から {1, 1, 0, 1}, {2, 0, 1, 0}, {1, 1, 2, 1} であり、要素は順に 1, 2, 3, 4 秒後に交差点に進入する車の台数を表している。つまり、このキューは図の矢印で示される順に車が移動していくことを表している。

A Traffic Simulation in Parallel and Its Evaluation  
Kenichi Tsukamoto, Yasuhiro Matsuoka, Hidenori Itoh and Hirohisa Seki  
Nagoya Institute of Technology  
Gokiso-cho, Showa-ku, Nagoya 466, Japan

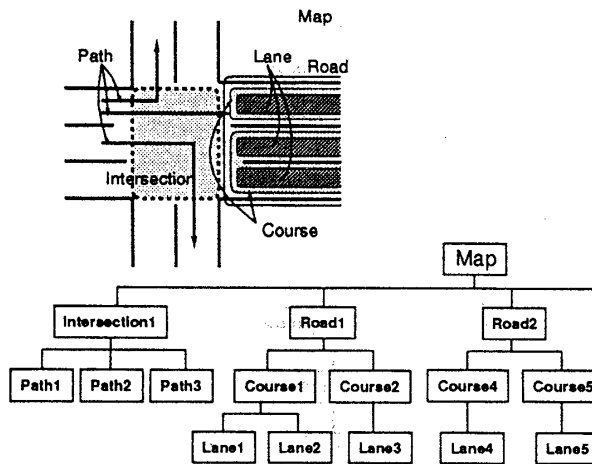


図1: 交通網のデータ構造

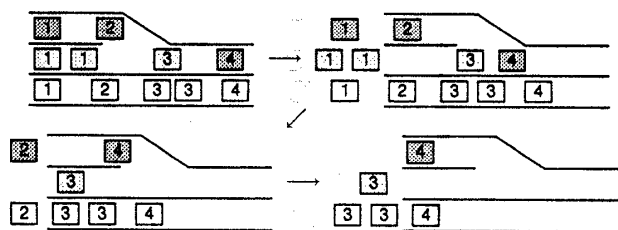


図2: Lane クラスの持つキュー

## 3 シミュレーションの並列処理

本システムを、メッセージ通信を基本とした分散メモリ型の並列計算機である AP1000 上にインプリメントした。

### 3.1 データの配布

本システムでは、各プロセッサに、サーバと呼ばれる一つのタスクと、クライアントと呼ばれる複数のタスクを生成する。交通網データベースは、シミュレーション処理を行う前に、各プロセッサあたりのオブジェクト数が同じになるように割り当てられ、サーバがこれを受け取る。

### 3.2 シミュレーションの本処理

シミュレーションは、計算処理を行うステップ1と、データの同期をとるステップ2からなる。

#### ■ ステップ1

オブジェクトごとの処理を各クライアントが行う。そのなかで、必要に応じてサーバとクライアントの間でメッセージ通信が行われる。図3において、円はプロセッサを、四角はタスクを表しており、矢印は Lane クラスのオブジェクト Lane1 の処理過程におけるメッ

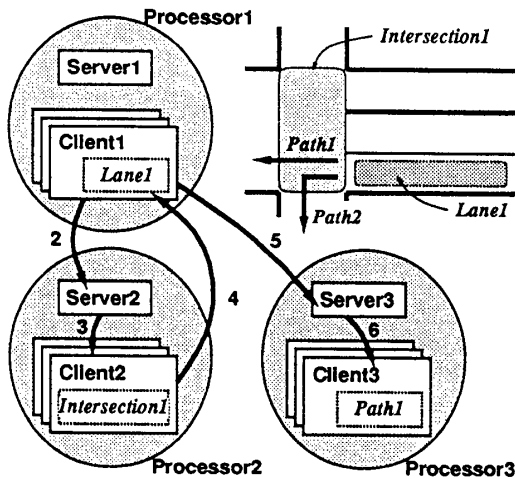


図3: タスク間のメッセージ通信

メッセージ通信を示している。

オブジェクト Lane1 の処理過程は次のようになる。番号は図3のメッセージ通信の番号に対応している。

1. Client1 では、キューの先頭に車があるかどうかを調べる。なければここで処理は終了する。
2. Lane1 には Path1, Path2 の2つのパスが接続している。Lane1 が接続している交差点 Intersection1 を処理しているのは Processor2 であるので、Client1 は Server2 に進入すべきパスを問い合わせる。
3. Server2 は、パスの選択を Client2 に依頼する。
4. Client2 は、決定したパス (ここでは Path1) を Client1 に知らせる。
5. Path1 を処理しているのは Processor3 であるので、Client1 は Server3 に対して、車の進入を知らせる。
6. Server3 は、車の進入処理を Client3 に依頼する。

■ ステップ 2

各サーバは、ステップ1の処理結果に従ってデータベースを更新する。上の例で取り上げた Lane1 の処理では、ステップ1で計算したパスへ進入する車の台数に従ってキューを更新する。

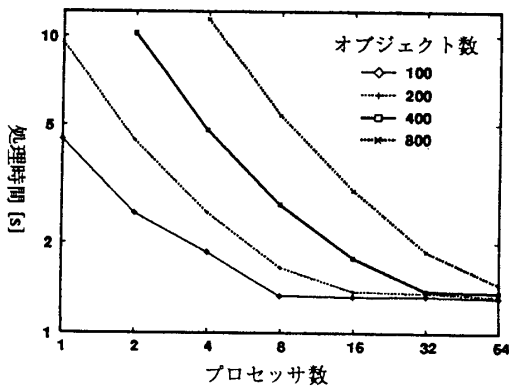


図4: 並列効果の評価

### 4 並列効果の評価

並列効果を評価するために、オブジェクト数の異なるデータベースについて処理時間を測定した。用いたデータベースは4種類で、それぞれ100, 200, 400, 800のオブジェクトから成っている。この実験の結果を図4に示した。プロセッサ数に対してオブジェクト数が少ない場合には、プロセッサ数を増やしても処理時間が少なくなりますが、オブジェクト数が充分多い場合には並列効果が出ることを確認した。

### 5 シミュレーション例

本システムを用いて、交通網を走っている車の台数と、一台の車が単位時間に進む距離およびすべて車が単位時間に進む距離の和との関係を図5に示す。実験環境のデータベースは、ノード数200、交差点数21、道路数24であり、これは中規模程度の都市の道路網に相当する。なお、プロセッサ64台を並列に使用した。

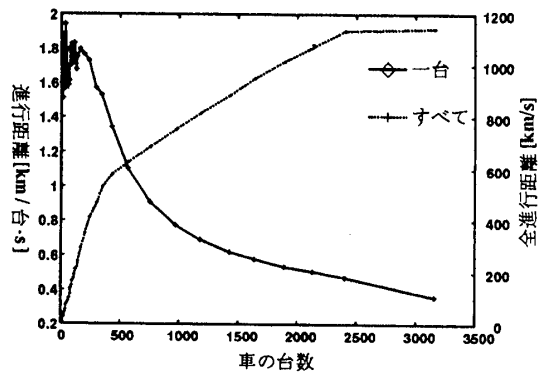


図5: 車の台数と進行距離との関係

車の台数が増加するとともに個々の車の移動距離が少なくなり、個々の車の移動距離の総和は一定となることが示された。

### 6 おわりに

交通流シミュレーションシステムを並列計算機上に実現し、評価を行った。今後は、オブジェクトの割当ての際に、交通網の地理的な条件を加味することによって、メッセージの通信量を押えることを検討する。

謝辞

本研究は計算機資源として富士通 AP1000 を使用した。また、佐川交通社会財団の助成により実施した。

参考文献

[1] Yasuhiro Matsuoka, Hidenori Itoh and Hirohisa Seki: A Traffic Simulation System and Its Parallel Implementation. Proceedings of 16th International Conference on Computers & Industrial Engineering, March 7-9, 1993.