大規模並列交通シミュレータの実現と負荷分散方式の評価

尾崎敦夫[†] 占市昌一[†] 阿部一裕[††]
中島克人[†] 田中秀俊[†]

渋滞解消のための実時間信号制御や，広域な交通網整備などの用途への活用を目指した大規模交通シミュレータを開発した．本シミュレータは提案した時空間オブジェクトモデルを基礎にインテル社製のMIMD型高並列計算機Paragon上に開発したものである．本シミュレータの性能評価の一環として，並列処理の単位となる粒度に関して，高性能を得る最適な粒度を求めるための負荷バランス方式の実験を行った．負荷バランス方式は，道路網を細かく分割して，各プロセッサへラウンドロビンに割り振る多重マッピング方式を採用した．横浜市の中心部4km × 2km四方の領域の実データを使用し，Paragonの32プロセッサを用いて，多重マッピング方式を適用した場合に，約3,000台の車と約1,400の信号付き交差点を実時間実行できることが分かった．このケースでは，負荷バランスを考慮しない単一マッピング方式と比べて約2倍の性能向上が達成できている．

Design of a Parallel Car Traffic Simulator and Its Evaluation for Load Balancing

Atsuo Ozaki,[†] Masakazu Furuichi,[†] Kazuyuki Abe,[††]
Katsuto Nakajima[†] and Hidetoshi Tanaka[†]

We developed a large scale car traffic simulator based on a Space-Time Object model. The target application of this simulator is real time traffic lights control, design of road network and so on. In this paper, we discuss issues in implementation and the performance evaluation of the simulator. We also present the results of static load balancing. Two mapping schemes have been applied for estimation of the performance of load balancing. One is a **one-to-one** mapping scheme, in which one sub-road is mapped onto one processor. Another is **multiple** mapping scheme, in which more than two sub-roads are mapped onto one processor. The simulator based on **multiple** mapping can simulate about 3,000 cars and 1,400 traffic lights at real time speed on 32 processors of Intel Paragon using the actual road map of a 4km × 2km area in the heart of Yokohama City. The performance of the simulator based on **multiple** mapping is about two times faster than that of **one-to-one** mapping in this case.

1. まえがき

交通車両の増加にともなう交通渋滞は，快適なドライブの妨げとなるだけでなく，排気ガスによる大気汚染や，騒音などの環境問題を引き起こす大きな要因となる．社会問題となって久しい交通渋滞を解決するための方法としては，交通状況を瞬時に反映できる実時間信号制御や旅行時間情報の提供，そして道路の拡幅などが考えられる．これらを実現するためのツール，または効果を検証するための手段として計算機を用いたシミュレーションが重要視されている．特に，渋滞などは部分的に解消できても意味がなく，また旅行時間情報の提供や道路拡幅などの交通網整備を行ううえでも，少くとも町村レベルまたは市レベルをターゲットにした大規模な交通シミュレータが必要となる．また，このような目的のためには渋滞などの系の挙動予測が困難であるような状況をも精度良く模擬する必要がある．このような系全体の挙動を定式化するのが困難である問題には，ミクロレベルを基本とするシミュレーションが適している．このミクロレベルシミュレーションは，個々の車オブジェクトの状態変化や車間の相互作用などを定義し，ダイレクトマッピング法[20]を基礎にプログラム化するものである．また，交通シミュレーションでは，車は近傍の車や信号などの情報だけ

† 三菱電機株式会社情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation
†† 三菱電機株式会社先端技術総合研究所
Advanced Technology R&D Center, Mitsubishi Electric Corporation

を必要とし，オブジェクト間通信はシミュレーション空間上で局所的に行われるため，多くの並行性を抽出することができるという特徴がある．

我々は，このミクロレベルの交通シミュレーションを並列計算機上で構築するための計算モデルとして時空間オブジェクトモデル[1),4),5),15),17),18)]を提案し，論理プロセスの枠組みを提供する並列オブジェクト指向シミュレーション環境OSim[1)]上に本計算モデルを実装した．本計算モデルは，車などの移動体オブジェクトの「時間」と「空間」を管理するField[15)]という概念を中核としており，オブジェクト間の通信はこのFieldを介した間接通信として実現される．具体的なFieldの機能としては，並行に動作するオブジェクト間で矛盾が生じないように時刻を管理する時刻管理機能と，オブジェクト間の位置関係を効率良く表現するための空間管理機能とがある．この時空間オブジェクトモデルは，オブジェクト指向モデルを基礎としているため，1つ1つの移動体を自律オブジェクトとして個性を持たせることが可能であり，また並列シミュレーション技術を導入しているため移動体の数が非常に多くなる場合でも計算量に比例してプロセッサの数を増やせば，要求に応じた性能を得ることができるという特徴がある．

本論文では，インテル社製のMIMD型高並列計算機Paragon上に構築した時空間オブジェクトモデルに基づく並列交通シミュレータ[7),18),19)]の実装方式と，Paragonのプロセッサを最大64台使用して実施した本シミュレータの性能評価について報告する．交通シミュレータはすでにいくつかのものが開発されており[12),14)]，並列計算機上に開発した大規模交通シミュレータとしては，ほかにSIMD型超並列計算機CM200上に開発したPARAMICS[6)]が存在するが，近年主流のMIMD型の並列計算機上に開発された大規模な交通シミュレータは存在しない．

以下，2章では，時空間オブジェクトモデルについて説明する．3章では，このモデルに基づく交通シミュレータの実装方式について述べる．4章では本シミュレータの基本性能の評価として行った台数効果について報告する．また，5章では問題サイズに応じて実時間実行を可能とするために必要となるプロセッサ台数と，並列処理の単位であり，負荷分散の単位でもある粒度の最適値に関して考察する．最後に6章でまとめと今後の課題について述べる．

2. 時空間オブジェクトモデル

交通シミュレーションを例題とする実世界シミュレーションでは，各移動体オブジェクトは，次のステップを繰り返すことによりシミュレーションが進行する．

step1： 次の時刻の状態を決めるため，近傍のオブジェクトと通信を行い，必要な情報を得る．

step2： step1で得た情報を基に次の時刻の状態を決定する．

この処理では，各オブジェクトは並行して模擬を行うため，オブジェクト間通信の際に通信相手のオブジェクトがどの時刻の状態であるかを管理する時刻管理機能が重要となる．またシミュレーション空間上の近傍オブジェクトを効率的に検索する空間管理機能も重要な要素となる．提案した時空間オブジェクトモデルではFieldの概念を中核とし，すべてのオブジェクトはこのField上にダイレクトマッピングされ，時間的にも空間的にもこのFieldにより管理される．

2.1 時刻管理

シミュレーションの時刻を進める単位を論理プロセス（LP: Logical Process）と呼ぶ[1)]．このLPは1つのオブジェクトであっても，またいくつかのオブジェクトを1つにまとめたものであってもよく，LP内のオブジェクトは同一時刻でシミュレーションが進行する．LP間の時刻に関しては，Time-Driven（TD）方式[9)]のようなすべてのLPが同一時刻となるように同期しながらシミュレーションを行うものと，Time-Warp（TW）方式[8),10)]やBreathing-Time-Buckets（BTB）方式[9)]，そしてBreathing-Space-Time-Buckets（BSTB）[11)]方式のような各LPで時刻を独立に進める方式[2),3)]が存在する．時刻を独立に進める方式では，シミュレーション時刻の異なるLP間で通信があった場合に因果関係の矛盾が生じる可能性があり，この矛盾を修復するために以下のような機能が必要となる．

- LPの時刻を巻き戻すロールバック処理機能
- ロールバックを行うために履歴を保存する機能
- ロールバック処理にともない，発送したメッセージを取り消すアンチメッセージ機能

2.2 空間管理

ミクロレベルの移動体シミュレーションでは，系全体が刻々と変化するため，各オブジェクトがstep1において通信を行う相手もシミュレーションを行う時刻とともに変化する．このためオブジェクト間通信はFieldを介した間接通信とした．この間接通信とは，オブジェクトがFieldに対して時刻と検索範囲を伝えることにより，Fieldはその時刻のその範囲に存在するオブジェクトを検出し，依頼元のオブジェクトに検出したオブジェクトへの通信経路を提供することにより可能となる通信である．

2.3 並列オブジェクト指向シミュレーション環境 OSim

我々はシミュレーション構築者であるユーザに，LP の枠組みを提供することを目的に開発した並列オブジェクト指向シミュレーション環境 OSim[1),11)] を並列交通シミュレータの開発に活用した．OSim は超並列オブジェクトベース言語 OCore[16)] 上のクラスライブラリとして実現しており，OCore の持つ以下のような諸機能を利用し，各種の時刻管理方式（TD，TW，BTB，BSTB）を効率的に実行できる環境を提供する．

- オブジェクトの集合を構造化し，通信の分散や効率的な実装を目的とする「共同体」機能
- 柔軟な同期を可能とする同期構造体

OSim が提供する LP は，時刻管理オブジェクトとアプリケーションオブジェクトより構成される（**図 1** 参照）．時刻管理オブジェクトは，LP の状態値の管理，メッセージキューの管理，そして LP 間の時刻同期を管理するオブジェクトであり，アプリケーションオブジェクトは，対象とするアプリケーションを実行するオブジェクトである．この各 LP は，並列計算機の各プロセッサに割り付けられ並列実行される．また，OSim では BSTB 方式以外は空間管理を行う機能を実現していないため，TD 方式や TW 方式などで時空間オブジェクトモデルを実現するためには，ユーザがプログラムを記述するアプリケーションオブジェクト内で空間管理を行う機能を構築する必要がある．

OSim の特徴は，適当な時刻管理オブジェクトの選定により，対象とするアプリケーションに適した時刻管理方式に基づくプログラムが実現できることである．

OSim では LP 間で送受信されるメッセージのうち次の 3 つをライブラリのインタフェースとして提供する．

Event 型メッセージ： LP の状態値を変更することができるメッセージ

Query 型メッセージ： LP の状態値を参照することができるメッセージ

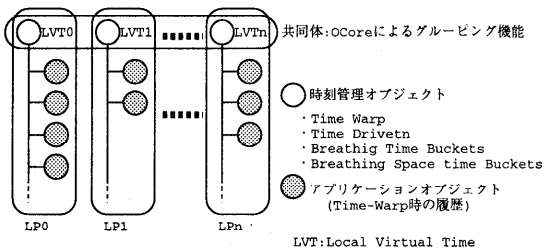


図 1 OSim の内部構造
Fig. 1 Structure of OSim.

Reply 型メッセージ： Query 型メッセージに対する返答のためのメッセージ

3. 時空間オブジェクトモデルに基づく並列交通シミュレータ

高並列計算機 Paragon 上に OSim を用いて時空間オブジェクトモデルに基づく並列交通シミュレータを構築した[7),19)]．以下にその実装方式，および対象としたシミュレーション領域と車の走行モデルについて述べる．

3.1 実装方式

交通シミュレーションでのシミュレーション空間すなわち Field に相当する道路網は，交差点をノード，道路をリンクとするグラフ構造で実現した（**図 2** 参照）．通信処理を頻繁に行う近傍のオブジェクト群は 1 つの LP としてシミュレーション時刻が同一に進むように実装するのが効率的である．このため，Field を複数の SubField（部分道路網）に分割し，各 SubField 内のオブジェクト群を LP の単位とした．そしてこの SubField を OSim のアプリケーションオブジェクト内に実装した．基本的に Field は，並列計算機が備えるプロセッサ数の SubField に分割して各プロセッサに 1 つの SubField が割り付けられる．

SubField 間の境界および境界付近には通信処理が多い交差点（ノード）が位置しないようにすべきである．このため，道路（リンク）に仮想交差点（VirtualIntersection）と称する 1 入力 1 出力のノードを挿入し，この仮想交差点を境にして SubField に

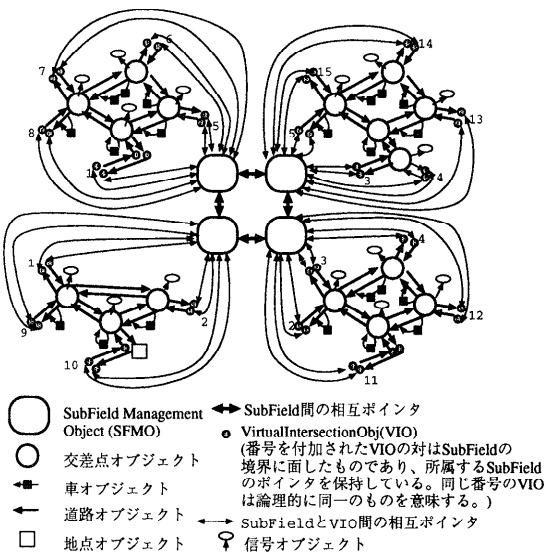


図 2 道路網の空間管理構造
Fig. 2 Structure of space management.

分割することにした（図2参照）．各SubFieldには，SubField内の車や信号の模擬をスケジュール管理するSubField Management Object（SFMO）が存在する．各SFMOは時刻を進めるためのEvent型メッセージを自分自身に送信することにより，シミュレーションが進行する．時刻進行のEvent型メッセージを受け取ったSFMOは1秒間分の模擬を車や信号に対して行うものとした．また，SubField間の車の移動にともなう通信処理はOSimが備えるEvent型，Query型，そしてReply型の各メッセージを用いて実現した（**図3**参照）．

信号オブジェクトは，交差点へ車が流入するすべての道路に割り付けた．また，すべての信号オブジェクトは同期しながら，青，黄，赤のサイクルを順番に状態遷移するようにした．ただし，各交差点で1サイクルに要するシミュレーション時間を同一にしたので，三差路と，十字路とでは，青，黄，赤の点灯時間は異なるものとなる．

車オブジェクトは道路または地点単位でリスト管理されている．地点とは，各車が出発する場所または目的地であり，警察署，病院，オフィスビル，消防署，そして住宅地がある．SFMOが時刻進行のためのEvent型メッセージを受信すると，以下の手順で各車は模擬される．

1. 進行方向の道路または地点の車リストから前方車の位置や速度などの情報を収集する．前方を見る範囲はその車の位置と速度から決まり，前方車の検索はその範囲まで道路リンクをたどることにより行う（図2参照）．
2. 前方車が検索できた場合は，式(1)により加減速度を計算し，次のシミュレーション時刻の位置と速度を決める．そして，計算した位置情報を基に該当する道路または地点の車リストに登録する．式(1)による車両追従モデルは，米国連邦道路庁（FHWA）によって開発されたミクロシミュレーションシステム「TRAF-NETSIM」において使用されたものであり，本モデルを基本とする交通シミュレータは渋滞現象などを高精度で模擬できるため渋滞解消のための信号制御などのツールとして非常に有効である[13]．

$$\ddot{X}_s(t+\Delta t) = c\frac{\dot{X}_f(t) - \dot{X}_s(t)}{\{X_f(t) - X_s(t)\}^m} \quad (1)$$

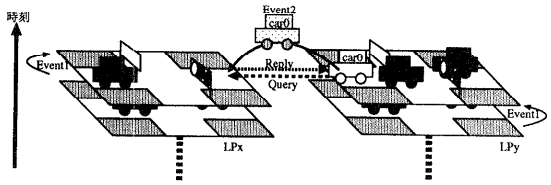
ここで，tは現在のシミュレーション時刻，Δtはシミュレーション時刻の刻み幅，Xは車の位置，sは模擬対象車，fはsの前方車，cとmは定数である．

本シミュレータでは，$c=1$，$m=1$とした場合の車の挙動が最も現実に即していたことからこの値を採用した．

また，前方のオブジェクトが信号であり，その状態が赤または黄である場合は，式(1)において$\dot{X}_f(t)=0$，$X_f(t)=0$とすることにより加減速度を決めるものとした．検索範囲に前方車または信号が存在しない場合は，その車が持つ加速度情報を基に加速するものとした．ただし，各道路には最高速度が割り付けられており，その速度は超えないものとした．前方を検索する範囲がSubField間を跨ぐ場合は，SFMO間でQuery型メッセージとReply型メッセージを交換することにより検索処理を行う．また，車がSubField間を移動する場合はEvent型メッセージを移動先のSFMOへ送ることにより実現した．

3.2 シミュレーション領域と車の走行モデル

道路網データとしては，横浜駅を中心とする6 km × 4 km区間のものを利用した．**図4**は，そのシミュレーション領域を示したものであり，この領域内では，



Event1:時刻を進めるためのEvent型メッセージ
Event2:車オブジェクトがLP間を移動するためのEvent型メッセージ
Query:隣接したLP内の状態を見るためのQuery型メッセージ
Reply:Query型メッセージの返答を行なうReply型メッセージ

図3 SubField内およびSubField間のメッセージのやりとり
Fig. 3 Message passing within a SubField and between SubFields.

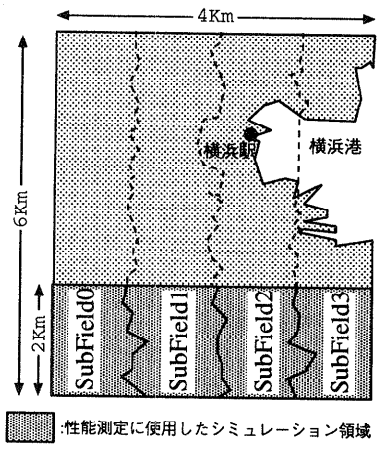


図4 シミュレーション領域
Fig. 4 Simulation area.

都市高速道路，一般国道，主要地方道（都道府県道，指定市道），一般都道府県道，そして指定市の一般市道などのすべての道路を実現した．SubField 間が直線で分けられていないのは境界線上に交差点が位置しないようにしたためである．この区間内に，交番，消防署，病院，住宅地，オフィスビルなどの合計 2,568 地点と，2,891 個の信号付き交差点を設定した．この中で，交番，消防署，そして病院は実際に存在する地点であるが，それ以外の住宅地やオフィスビルは横浜市の都市計画部が作成した「横浜市用途地域指定替素案」を基に各々の地点を人為的に設定した．なお，信号機は交差点へ車が流入するすべての道路に割り付けたため，実際の信号機数よりかなり多いものとなっている．各地点からは，目的地を与えられた車が発生するようにしており，発生した車は各々の目的地に向かって，道路の状態（制限速度，最高速度など），信号，そして前方車を見ながら加減速処理を行い走行するように設定した．

4. 性能評価

性能評価では Paragon のプロセッサ（PE: Processing Element）を最大 64 台使用した．**表 1**，**表 2** は，実験に使用した Paragon の PE およびネットワーク性能である．

実験では，初期状態時の車の台数が各 SubField でほぼ同じになるように地点の密度（地点数/m^2）が均一になる領域を選出した．実験に使用したシミュレーション領域は，図 4 に示す 4 km × 2 km 区間の横浜市中心部であり，総地点数は 1,129，信号付き交差点数は 1,400 である．領域の中心部はオフィスビル街，周辺部は住宅地域になっている．各車の行動モデルは，住宅地から発生した車はランダムに選ばれたオフィスビルを目的地に，またオフィスビルから発生した車はランダムに選ばれた発生地点以外のオフィスビルを目的地に設定して走行するようにした．また，道路上に車が存在しない状態を初期状態とした．このため，車の行動モデルは**図 5** に示すようにシミュレーション領域の中心部に車が集まる「集中型モデル」となる．集中型モデルは朝の通勤時に相当するものであり，車が一部の領域に集中するため，領域分割による並列処理ではプロセッサ間の負荷のバランスを崩す代表例である．このため，各種の負荷バランス方式の効果を評価するには適したモデルであると考える．

初段階として，TD 方式による並列交通シミュレータの性能評価を実施した．この結果，Paragon 1 PE では約 1,000 台の車を実時間実行できることが分かった．この実時間実行とは，たとえば 1 時間分のシミュレーションを 1 時間以内に計算機で処理することができることを示す．SFMO が時刻を進めるための Event 型メッセージを受け取るごとにすべての車に対して 1 秒間分の模擬を行うため，1 台の車の模擬に要する実行時間は約 1 ミリ秒ということになる．

また車の台数が case A（1,129 台），case B（2,258 台），case C（3,387 台），そして case D（4,516 台）の各ケースについてスピードアップを計測した（**図 6** 参照）．この結果，4 PE を用いたときの性能を 1 として，32 PE を使用した場合との比率は，case A が 6.8 倍で最良となる．これに対し，32 PE を用いたときの性能を 1 として，64 PE を使用した場合との比率は，case C が 2.9 倍と最も良く，case A が 1.3 倍と最も悪くなることが分かった．ちなみに，プロセッサ数を

表 1 Paragon の PE 性能
Table 1 PE performance of Paragon.

PE	i860XP（50 MHz）
性能	100 Mflops（単精度），75 Mflops（倍精度）
メモリ容量	16 Mbyte

表 2 Paragon のネットワーク性能
Table 2 Network performance of Paragon.

バンド幅	94 Mbyte/sec
通信遅延	38 μsec

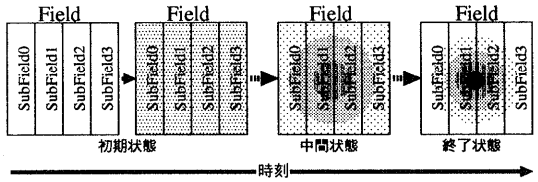


図 5 集中型モデル
Fig. 5 Concentration-type model.

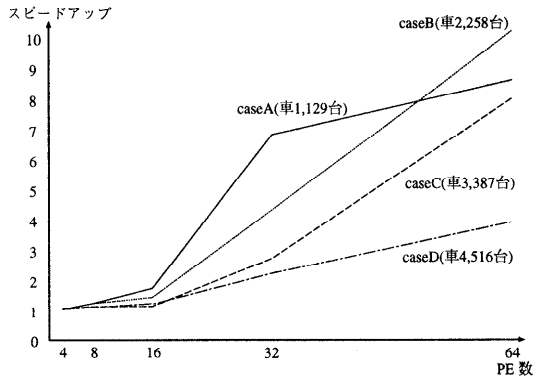


図 6 問題サイズごとのスピードアップ
Fig. 6 Speed-up from 4 to 64 PEs for each problem size.

倍に増やして性能が2倍以上になるのは，プロセッサ数を増やすとメモリの総量が増えるために2次記憶までデータをとりに行く回数および量が少くなるためである．これらの結果から，高いスピードアップを得るには，各プロセッサにおける負荷（車台数）とプロセッサ間の通信量の割合が重要な要素となることが分かる．

評価を行った車台数の妥当性に関しては，たとえばcase Dの場合において各車の車長を一律5メートルとすると，車の道路総占有長（22,580メートル）は，評価対象の4km × 2km区間に存在する道路の総延長（約183,209メートル）の約12%強を占めるものであるため，シミュレーション領域の中心部（オフィスビル街）では渋滞が発生する現実に近いものであると考える．表示系を用いた動作検証では，部分部分で適度な混雑状態が模擬できており，特にオフィスビル街においては渋滞現象も確認できている．

5. 考　　察

本章では，問題サイズに応じて実時間シミュレーションを実現するのに必要となるプロセッサ台数と，性能を向上させるための最適な粒度について考察する．

5.1 実時間シミュレーションに必要なプロセッサ台数の検討

本節では，問題サイズ（車の総台数）が与えられた場合に，何台のプロセッサを用いれば実時間実行が可能となるかを考察する．

TD方式によるシミュレータの実行時間は，最も実行時間が遅いPEの実行時間と等価であるため，シミュレータの実行時間 f は式(2)により与えられる．ここで f は，隣接PEと通信を行うのに要する実行時間 $g \cdot \sqrt{n}$ と，PE内に閉じた処理に要する実行時間 h の和である．

$$f = g(\rho) \cdot \sqrt{n} + h(\rho) \qquad \rho = N/n \tag{2}$$

g と h は，車の密度 $\rho = N/n$ に比例する．ここで，N は問題サイズであり，n はPE数である．Fieldは2次元構造であるのでSubField間の境界の長さは $\sqrt{n}$ に比例する．このため，密度が一定である場合の通信のコストは $\sqrt{n}$ に比例する．

図7は密度を各々一定にしたときの，PE数に対する150秒間分の模擬に要する実行時間を示したものである．図7の中の極細破線は，実際に計測した実行時間から式(2)の g と h の値を求め，PE数をこの式に代入することにより求めた f をプロットしたものである．この結果，たとえば密度（ρ）を62.5とすると52PEを使用することにより約3,000台（$62.5 \times 52 = 3{,}250$）の車を実時間実行できることが分かる（図7参照）．また図7と式(2)より，$\rho = 31.25$ とすると900PEを使用することにより10,000台以上（$31.25 \times 900 = 28{,}125$）の車を実時間実行できることが分かる．しかし，$\rho = 250$ または $\rho = 125$ とすると，6PEまたは12PEを使って1,500台の車しか実時間実行できない．このような解析から，問題サイズが与えられたときにPEをいくつ使えば実時間実行が達成できるかを見積もることができる．

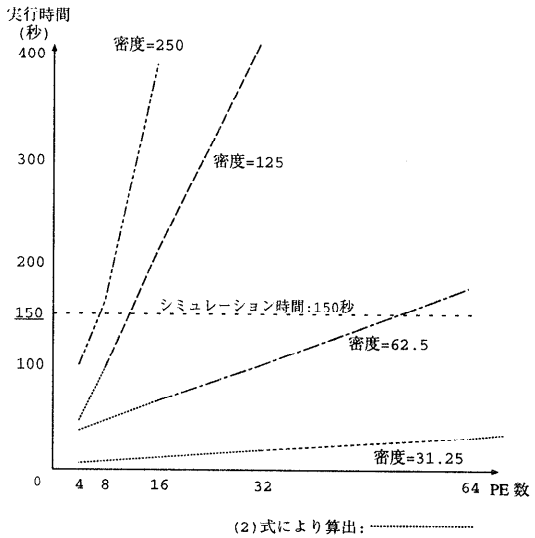


図7　密度を各々一定にした場合の150秒間分の模擬に要する実行時間

Fig. 7　Execution time vs. Number of PEs for 150 second steps when density is fixed in each case.

また，以上の結果から，64PEを備えるParagonでは，case C規模の車台数までを実時間実行できることが分かる．4km × 2km区間でのcase Dの車両密度と同等の状況を考えるのであれば，3km × 2km区間の領域が実時間実行を実現できる対象領域となる．これは小規模な町村レベルに相当するもので，現状では十数程度の信号機が協調制御されていることを考えると，このレベルでの信号制御はかなり規模の大きなものといえる．現状では，この協調制御している領域と領域の境界において渋滞が発生してしまうことや，この領域の流入出部分で必ず車が赤信号に捕まってしまうなどの問題があるが，このレベルでの信号協調制御が可能になると，これらの問題を大幅に解消することができる．

5.2 最適な粒度の検討

ここでは，SubField内の車の数を“粒度”と定義し，

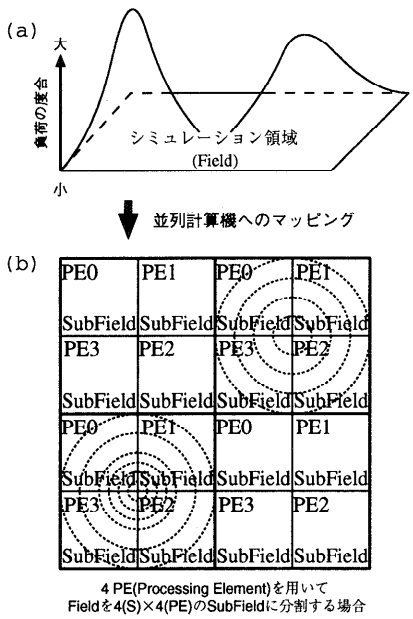


図8 多重マッピング方式
Fig. 8 Multiple mapping scheme.

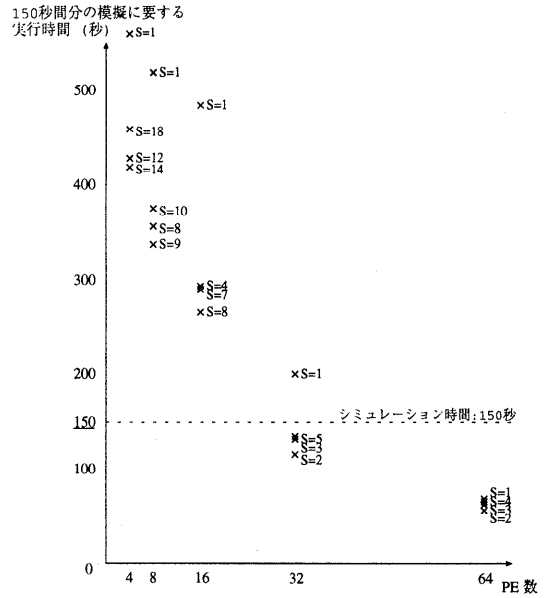


図9 case Cにおける最適な粒度と単一マッピング方式との比較
Fig. 9 Execution time for the best three grain size and one-to-one mapping scheme when the problem size is case C.

問題サイズが与えられたときに，粒度をどのくらいに設定すれば最高性能を引き出すことができるかを考察する．

図6は，1PEに1つのSubFieldを割り付けた場合の結果である．この方式を**単一マッピング方式**と称する．単一マッピング方式では，各SubFieldで車の数が異なるためにPE間で負荷のばらつきが生じる．この負荷のばらつきは全体の性能を落とす大きな要因となる．

この問題を解決する方法の1つとして，FieldをPE数 × S 個のSubFieldに分割して，各PEに S 個のSubFieldを割り当てる方式がある．この方式を**多重マッピング方式**と称する[18]．ここで $S \geq 1$ とする（$S = 1$ の場合は単一マッピング方式を示す）．同じPEに割り付けられるSubFieldは，シミュレーション空間上において互いに離れている領域を担当することが好ましい．これは，負荷の高い領域（車が密集している部分）または低い領域を，各PEに分散できるからである．**図8**は $S = 4$ として，4PEにより多重マッピング方式を適用した場合の例であり，各PEの負荷が均一となる状態を示している．

図9はcase Cにおいて，各PE数ごとに最も高い性能が得られた上位3つの S の，150秒間分の模擬に要する実行時間を示したものである．また単一マッピング方式との比較のために，$S = 1$ のときの実行時間もプロットした．この結果から，多重マッピング方式を適用すると，単一マッピング方式より最高で2倍近い性能が得られることが分かる．

また図9より，case Cにおける最適なSubField数（$S \times$ PE数）は，ほぼ70前後と考えることができる．この値を用いると，case Cにおける最適な粒度は，48（$\simeq 3{,}387/70$）となる．同様に，case Aの場合の最適な粒度は28であり，case Bの場合は38であった．問題サイズが大きくなると最適な粒度も大きくなる．これは，SubFieldの数が多くなるとSubField間の通信が増え，PE間の通信コストが高くなると同時に，PE内においてもSubFieldのスケジューリングオーバヘッドが高くなるためである．このため，問題サイズが大きくなると，粒度も比例して大きくする必要がある．ここで得られた結果は，問題サイズが与えられたときに最適な粒度を選ぶ目安とすることができる．

6. むすび

本論文では我々が提案した「時間」と「空間」を管理する時空間オブジェクトモデルを基本に開発した並列交通シミュレータの実装方式と，64プロセッサを備える高並列計算機Paragonを使用して実施したTD方式による並列交通シミュレータの性能評価に関して報告した．この性能評価の結果，本交通シミュレータを実時間信号制御などに使用する場合には，小規模な町村レベルまでを扱えることが分かった．

また，問題サイズが与えられたときに，実時間シミュレーションを実現するために必要なプロセッサ台数や，高性能を得るための最適な粒度に関して考察した．この結果，高性能を得るためにはSubField間の通信処理とSubField内の処理との比率が重要であると同時に，負荷がすべてのプロセッサへ均等に分散できるような適当な粒度の選定が重要であることが分かった．特に道路交通のような，シミュレーション空間において負荷が高いところと低いところが極端となる応用分野では，負荷を各プロセッサへできるだけ均等に分散させることが性能を高めるための重要な鍵となる．またシミュレーション時刻の進行にともない，負荷の高い部分（ホットスポット）がシミュレーション領域を移動するためにプロセッサ間で負荷の不均等が生じ，性能を悪化させてしまうような場合では，シミュレーション中に負荷をプロセッサ間で均等にするような処理が必要となる．1つの方法として，シミュレーション中にSubFieldなどの処理単位をプロセッサ間で移動させ，負荷を均等化する動的負荷バランス方式が存在する．しかし，動的負荷バランス方式は負荷が不均一になる状態を検出するための機能が必要となると同時に，SubFieldをプロセッサ間で移動させるコストが非常に高くなるために非現実的である．本論文で述べた静的負荷バランスを基本とする多重マッピング方式は，負荷が不均等であり，ホットスポットがシミュレーション空間を移動するような場合においても負荷を適度に均等化できるという特徴がある．

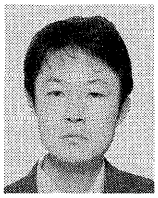
今後の課題は，TD方式以外の時刻管理方式を基本とするシミュレータの性能評価と，各々のシミュレータ間の比較評価を行うことである．また，本交通シミュレータはOCoreという高位の並列オブジェクト指向言語を用いて開発しているため，さらに効率の良い並列プログラム実行環境を用いることにより，1桁程度性能を向上できると考えられる．我々は，この改良作業と多重マッピング方式などの最適化を施すことにより市レベルの領域を実時間実行の対象領域とすることができると考えており，これらを検証することも今後の課題である．

参考文献

1) Abe, K., et al.: A Massively Parallel Object Oriented Simulation Environment OSim, *Workshop on Object-Oriented Computing 96*, S3-2, No.2, ISSN 1341-870X (1996).
2) Fujimoto, R.: Parallel Discrete Event Simulation, *Comm. ACM*, Vol.33, No.10, pp.30–53 (1990).
3) Fujimoto, R.: Parallel and Distributed Discrete Event Simulation: Algorithms and Applications, *Proc. 1993 Winter Simulation Conference*, pp.106–114 (1993).
4) Furuichi, M., et al.: A Space-time Object: An Object Oriented Model for Parallel Simulation, *Proc. Object Oriented Simulation Conference*, pp.86–91 (1996).
5) Furuichi, M., et al.: A Space-Time Object – Parallel Object Oriented Simulation Environment and its Evaluation, 1997 Real World Computing Symposium Proceedings, RWC Technical Report (TR-96001), pp.513–520 (1997).
6) Cameron, B.G. and Wylie, D.M.: PARAMICS – Moving Vehicles on the Connection Machine, *Super Computing '94*, pp.291–300 (1994).
7) Ozaki, A., et al.: Parallel Car Traffic Simulation Based on Space-Time Object, *ESS96 (8th European Simulation Symposium)*, Vol.II, pp.33–37 (1996).
8) Presley, M.T., et al.: A Time Warp Implementation of Shark's World, *Proc. 1990 Winter Simulation Conference*, pp.199–203 (1990).
9) Steinman, J., et al.: SPEEDS: A Multiple-Synchronization Environment for Parallel Discrete-Event Simulation, *the International Journal for Computer Simulation*, Vol.2, No.3, pp.251–286 (1992).
10) Steinman, J., et al.: Breathing Time Warp, *Proc. 7th Workshop on Parallel and Distributed Simulation (PADS93)*, Vol.23-1, pp.109–118 (1993).
11) 阿部一裕ほか：空間的影響範囲を考慮した分散論理時刻管理方式，情報処理学会論文誌，Vol.40, No.3, pp.1018–1026 (1999).
12) 滑川光裕ほか：道路交通ネットワークを対象とする並列シミュレーションのための同期アルゴリズム，第16回シミュレーション・テクノロジー・コンファレンス発表論文集，pp.185–188 (1997).
13) 吉川康雄ほか：渋滞改善のための交通流ミクロシミュレータの開発，第14回シミュレーション・テクノロジー・コンファレンス発表論文集，pp.189–192 (1995).
14) 後藤幸夫ほか：自律型走行モデルによる道路交通シミュレータの開発，電気学会論文誌D，116.5 (1996).
15) 小中裕喜ほか：並列計算機上のオブジェクト間の間接的通信の実現について，情報処理学会プログラミング—言語・基礎・実践—研究会報告，Vol.9-3, pp.17–24 (1992).
16) 小中裕喜ほか：超並列オブジェクトベース言語OCoreの商用並列計算機上での実装，並列シンポジウムJSPP，pp.113–120 (1994).

17) 尾崎敦夫ほか：時空間オブジェクトモデルの基本設計—ミクロ交通シミュレーションへの適用検討，第47回情報処理学会全国大会論文集，Vol.1B-2 (1993).
18) 尾崎敦夫ほか：時空間オブジェクトモデルを用いた並列シミュレーション—ミクロ交通シミュレーションへの適用検討，第13回シミュレーション・テクノロジー・コンファレンス発表論文集，pp.299–302 (1994).
19) 尾崎敦夫ほか：オブジェクト指向並列シミュレーション技法に基づく交通シミュレータの実装と評価，第15回シミュレーション・テクノロジー・コンファレンス発表論文集，pp.173–176 (1996).
20) 村岡洋一：超並列マシン—その新しいプログラミングスタイル，*Computer Today*, Vol.9, No.49, pp.27–34 (1992).

(平成10年4月1日受付)
(平成11年3月5日採録)

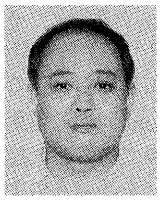


尾崎　敦夫（正会員）

1964年生．1988年九州工業大学工学部情報工学科卒業，1990年同大学院電気工学部計算機コース博士課程前期修了．同年4月より三菱電機（株）情報電子研究所に勤務．1992年から1996年までリアルワールド・コンピューティング（RWC）プロジェクトに参画し，並列シミュレーションの研究開発に従事．現在同社情報技術総合研究所において，並列分散シミュレーションシステムの研究開発に従事．負荷バランス，オブジェクト指向シミュレーション，並列分散協調型シミュレーションに興味を持つ．1996年 European Simulation Symposium Best Paper Award 受賞．

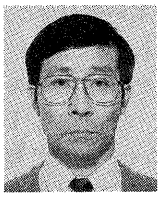
古市　昌一（正会員）

1958年生．1982年広島大学総合科学部総合科学科卒業．同年4月より三菱電機（株）情報電子研究所に勤務．1992～1994年イリノイ大学アーバナシャンペン校コンピュータサイエンス学科大学院に留学，修士課程修了．現在三菱電機（株）情報技術総合研究所において，並列分散シミュレーションシステムの研究開発に従事．負荷バランス，オブジェクト指向シミュレーション，分散協調型シミュレーションに興味を持つ．ACM，IEEE Computer Society，米国シミュレーション学会（SCS）各会員．



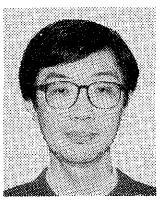
阿部　一裕（正会員）

1965年生．1988年大阪大学工学部応用物理学科卒業，1990年同大学院応用物理学コース博士前期課程修了．同年4月より三菱電機（株）中央研究所に勤務．1992年度から1996年度までリアルワールド・コンピューティング（RWC）プロジェクトに参画し，並列分散シミュレーション方式の研究開発に従事．現在同社先端技術総合研究所において，ネットワークエージェントシステムの研究開発に従事．モバイルエージェント，分散協調処理方式に興味を持つ．



中島　克人（正会員）

1953年生．1977年京都大学工学部電気第二工学科卒業，1979年同大学院修士課程修了．同年三菱電機（株）に入社し，汎用・専用計算機の開発に従事．1982年より第五世代コンピュータ・プロジェクトに参画し，逐次型および並列型推論マシンのアーキテクチャ/言語処理系等の研究開発に従事．1993年よりリアルワールド・コンピューティング（RWC）プロジェクトに参画．現在，同社情報技術総合研究所において，並列・分散処理向けアーキテクチャおよびミドルウェアの研究開発等に従事．最適設計・スケジューリング技術等にも興味を持つ．工学博士．IEEE Computer Society 各会員．



田中　秀俊（正会員）

1963年生．1986年東京大学工学部計数工学科卒業．同年4月三菱電機（株）入社．1989～1995年（財）新世代コンピュータ技術開発機構に出向，現在三菱電機（株）情報技術総合研究所において，主にデータ解析および最適化に関する研究開発に従事．IEEE Computer Society 各会員．