

マルチプロセッサ型交通シミュレータの同期化手法

五十嵐 智也[†] 中村俊一郎[†] 宮西 洋太郎[‡] 吉田 幸二[£] 齋藤 成一[¶]
 日本工業大学[†] 公立はこだて未来大学[‡] 倉敷芸術科学大学[£] 三菱電機(株)[¶]

1. はじめに

ここ数十年、都市およびその近郊における交通渋滞は一向に解消されていない。交通の円滑化を図るためパソコンを用いたアプローチも多く、入手可能なシミュレータも多数存在する^[1]。しかし独自機能を盛り込むためわれわれは独自の交通シミュレータを開発している。2000年度にはスタンドアロンの交通シミュレータを開発。昨年度はそのシミュレータのマルチプロセッサ化を行い、発表を行った^[2]。今年度はこれに各パソコンの動作クロックの完全同期機能を付与したので、その報告を行う。

2. 広域画面表示の問題点

交通シミュレータの代表的な出力方法はアニメーション画面、統計情報(累積交通量、平均車頭間隔、その他)の2つである。前者のほうが視覚を通し、容易に状況をつかむことが可能だろう。また、処理方式という点でもリアルタイムシミュレーションとノンリアルタイムシミュレーションの2つに分けることができる。前者のほうが処理に時間がかかるといった短所があるものの、任意の変更が随時可能な点が後者よりも優れている。具体的な例では、図1のシミュレーション実行中に上馬交差点の信号のタイミングを変化させ、大原交差点にどのような影響を与えるか(+の影響、-の影響共に)ということが、アニメーション画面のリアルタイムシミュレーションでは容易に確認することが可能であろう。

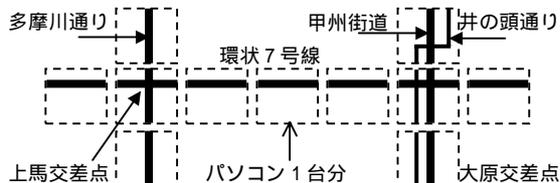


図1 広域シミュレーション処理

近年のコンピュータが高性能化し、マイクロシミュレータ^[3]でも比較的広範囲のシミュレーションを行うことが可能となってきた^[4]。しかしこれは計算が可能、という意味でありアニメーション表示のリアルタイムシミュレーションで認識が可能ということの意味しない。われわれの研究で、シミュレーション領域を1000m * 700m、最小要素である一台の車の表示は2 * 4ドットが認識できる最小サイズであると結論付けた。この範囲を拡大するためにはシミュレーション領域を拡大し、その中で見たい部分に対してズームを行い、画面をずらしていくといった虫眼鏡で見ていくような方式が採られがちである。しかしこの方法では先ほどの例のように、ある地点で変化を与え、その変化が他に影響していく様子を観察するには適していない。

そこで我々は図1のようにパソコン一台分のシミュレーション領域は変えずに、複数のパソコンを接続し並列にシミュレーションを実行させることにより広範囲なシミュレーションを行うことを昨年提案した。設置例を図2に示す。この方式のメリットは矩形領域以外でもフレキシブルにシミュレーション範囲を増減させることが可能という点で、利用するパソコンの台数を減らすことができるという利点もある。

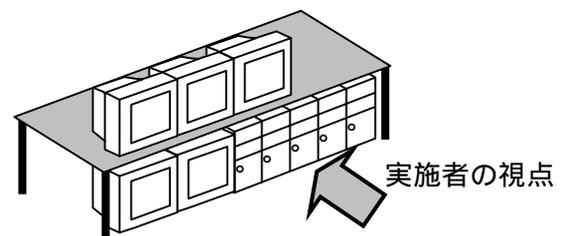


図2 実際のパソコン設置例

各パソコンは100M/bpsのイーサネットで接続し、安定性より速度を重視したUDP/IPをプロトコルに採用した。

3. 同期化手法

3.1 従来の方式

昨年発表した段階での我々のシミュレータの同期手法を図3に示す。Mはマスター、S

A synchronization method for multiprocessing traffic simulator

[†] Tomoya Igarashi, Shunichiro Nakamura

[‡] Nippon Institute of Technology

[‡] Yohtro Miyanishi [‡] Future University Hakodate

[£] Kouichi Yoshida [£] Kurashiki University of science and the arts

[¶] Seiich Saito

[¶] Mitsubishi Electric Corp

はスレーブプロセッサを示す。

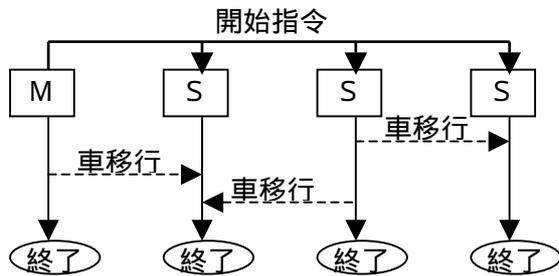


図3 開始時間同期方式(従来型)

この方式だと、開始時に一斉に指令をかけ同期させるが、その後プロセッサ間での同期を取ることとはしない。

3.2 今回の同期化方式

そのため極端に負荷が違う状況が起きた場合に同期することが困難な状況となっていた。そこで図4に示す完全同期方式を新たに作成した。

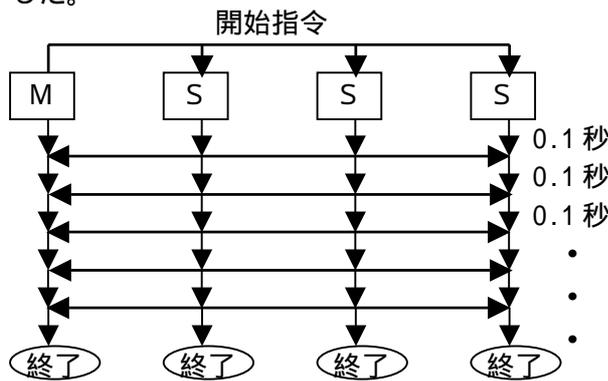


図4 完全同期方式

この方式では本シミュレータの最小時間単位0.1秒ごとに、シミュレーション開始指令をマスタープロセッサが送り、0.1秒分のシミュレーションを終えた各プロセッサはマスタープロセッサにその報告をする。マスタープロセッサはすべてのプロセッサから報告が帰ってきた時点でさらに、0.1秒分の処理を行う指示を出す。しかしこのように一回のシミュレーションごとに停止、開始を知らせる通信を行うため今までの処理よりかなりの処理速度低下が懸念された。結果を表1に示す。

表1 5分の処理にかかった時間

	マスター	スレーブ
開始時間同期方式	5:28	5:00
完全同期方式	5:28	5:28

表1はそれぞれの方式で5分の処理にかかった実時間である。開始時間同期方式ではマスタープロセッサとスレーブプロセッサの差が28秒

もあることがわかる。これだけの差が発生していると再現性を求めるのは難しいと思われる。当然ではあるが完全同期方式はマスター、スレーブともに同処理時間でシミュレーションが完了している。また開始時間同期方式のマスターと差がない処理時間で完全同期方式の処理が終わっている。このことから、完全同期方式でシミュレーションを行なってもシミュレーション性能の低下を招かないことがわかる。完全同期方式が開始時間同期方式とほぼ同じ処理速度というこの結果は、我々の予想以上のものであった。

この完全同期方式により各プロセッサの処理速度などに左右されない再現性を得ることが可能となった。開始時間同期方式では再現性に重きを置いていなかったために、同じ条件でシミュレーションを実行しても、完全に同じ状況になることは難しかった。またこの再現性によりマルチプロセッサ環境における早送りが可能となったと考えられる。開始時間同期方式では早送り処理が終わったプロセッサから順次、画面が表示されシミュレーションしていくという状況で、早送り中の時間が同期していないため、車が行進していくタイミングがまちまちだった。そのため開始時間同期方式では早送りはサポートしていなかった。この完全同期方式導入により今後早送り処理の開発に着手する基盤ができた。

4. 終わりに

昨年の開始時間同期方式をさらに発展させ、新たに完全同期方式を採用することができた。この方式により再現性の向上、早送り機能の実現の可能性が出てきた。また完全同期と言うオーバーヘッドが大きいと考えられる方式でも過去の開始時間同期方式とほとんど変わらない処理速度でシミュレーションを実行できることが確認できた。

参考文献

- [1] 堀口良太他：年街路網の交通流シミュレータ - AVENUE - の開発、第13回交通工学研究発表会、1993
- [2] 五十嵐智也他：マルチプロセッサ型交通シミュレータの開発、情報処理学会、情報処理学会第64回全国大会、2002
- [3] 猪飼國夫他：ファジィモデルに基づく市街地での渋滞予測用微視的シミュレータ、日本ファジィ学会誌、Vol.11、No.2、1999
- [4] 桑原雅夫：広域ネットワーク交通流シミュレーション、自動車技術、Vol.52、No.1、1998