Clock Synchronization Algorithm for Parallel Road Traffic Simulation

Mitsuhiro NAMEKAWA[1], Akira SATOH[2], Hideki MORI[2]
Kunio YIKAI[3], and Toshio NAKANISHI[4]

1) Kaetsu College 2) Toyo Univ. 3) MI Venture's Co. 4) Seikei Univ.

ABSTRACT Parallel simulation has contributed to reducing the execution time in simulating a large complex system. The system is partitioned into N disjoint subsystems, each of which is assigned to one of N processors and the simulations of subsystems are carried out using them in parallel. Parallel simulation needs more overhead time for simulation in comparison with a single processor. This is generated to synchronize the simulation clock and to process the communication between N distributed processors. Almost all parallel simulations have been conducted on systems which have been represented only through the discrete-change model. A few algorithms have been proposed for simulation using the continuous-change model. However, an universal and effective algorithm has not yet been established for parallel simulation with the continuous or the combined model. Thus, we studied on an algorithm to synchronize the simulation clock and a methodology for a communication based on road-traffic system simulation with the aim of establishing universal methods. Road-traffic simulation using a microscopic model includes both the discrete-change model and continuous-change model. We propose an algorithm which is based on the CTW(Conservative Time Window) approach and which predicts the simulation clock allowing synchronization between a subsystem and the neighboring subsystems. This paper describes a new algorithm and a methodology to synchronize the simulation clock in parallel. We successfully carried out parallel simulations through our algorithm and methodology, and verified their effectiveness.

1. INTRODUCTION

Simulations that target wide-area systems as in traffic and railroad control require a huge amount of calculations and considerable time to perform processing. Moreover, as the need for highly accurate simulation models grows, the amount of calculations has been increasing by about 10 to 30 times. If we therefore consider the use of simulation in real time, the need arises for high-speed processing that can reduce typical processing time by factors of from several tens to several hundreds. One method that is said to be effective in responding to this need is simulation by multiple processors arranged in a parallel.

Parallel processing features division of the target area into multiple subareas and allocation of one subarea to each processor, with all processors working simultaneously. A problem that arises here is the extent to which each processor can perform its processing independently.

Parallel simulation schemes differ from system to system; no general solution method exists and the one adopted is usually chosen because of its affinity with the system in question. Research related to such schemes normally targets discrete-event simulation, and we in particular are researching continuous-discrete-event simulation. This field up to now has not received much attention.

Therefore, we propose an algorithm for a new parallel simulation scheme in continuous-discrete-event simulation targeting a road traffic system. In this system, the area targeted for simulation is divided, which means that roads themselves will be divided into subareas. As a result, some processes like information collection and vehicle-transfer must be performed between processors, and the overhead processing required here must be taken into account.

Parallel simulation based on a discrete event model fundamentally requires synchronous processing every simulation clock cycle. However, we propose a method for omitting such synchronous processing between processors as much as possible. We also propose a new method for acquiring information by road network disconnect.

2. METHODS IN PARALLEL SIMULATION

Nearly all parallel simulations have been conducted on systems described by the discrete-change model. For parallel simulations using the discrete-change model, two main paradigms, the so-called optimistic and conservative, have been proposed. The optimistic paradigm requires both time and space for saving state variables and rollback. In this approach the simulation is carried out independently and locally in each processor and one processor's simulation clock is often different from those of the other processors.

On the other hand, the conservative paradigm is generally vulnerable to deadlock and memory overflow. And although the rollback process is not necessary in the conservative approach, it is necessary to synchronize the processors using some kind of method. It is also very important to minimize the numbers of times the processors are synchronized. The most suitable conservative method is one that makes it possible to decrease the number of times synchronization occurs and makes it possible to reduce overhead time.

Various optimistic approaches have been proposed since 1985 (Jefferson, etc.). The conservative ones have been proposed since 1979 (Chandy and Misra, etc.). Lately, Rassul Ayani and Hassan Rajaei have proposed a parallel simulation scheme which employs Conservative Time Windows (CTW). The system simulated is partitioned into N-disjoint subsystems, each of which is represented by an object. The scheme identifies a time window for each object such that events within these windows are independent and can be processed concurrently. Different windows may have different sizes if the nodes advance heterogeneously. Unlike in similar methods [Lubachevsky (1989), Nicol (1991)], the windows are not bounded by a global ceiling. The CTW-algorithm for parallel simulation is thus, in many cases, able to process faster than many good sequential methods.

3. FEATURES OF ROAD TRAFFIC SIMULATION

The three features of our road traffic simulation are below.

3.1 Continuous- and Discrete-change Model

In order to simulate a congestion of road traffic system, it is indispensable to describe vehicles having own their decision-making capabilities, and to carry out the simulation continuously with a 0.1-second simulation clock. Therefore, a microscopic continuous-change model is needed to describe the vehicle running. On the other hand, the signals in intersections are described by the discrete-change model. Thus the simulation for analyzing congestion in a road traffic system consists of a continuous- and a discrete-change model.

3.2 Clock Synchronization

The road network to be simulated is partitioned into N disjoint subareas, each of which is assigned to one of N processors, and the simulation is carried out by using the processors in parallel.

Figure 1 shows that a road network system to be simulated is partitioned into three disjoint subareas in parallel simulation.

Unavoidable overhead time is associated with the following synchronizations of the simulation clocks among the distributed processors.

(a) the transfer of vehicles from one subarea to another subarea.

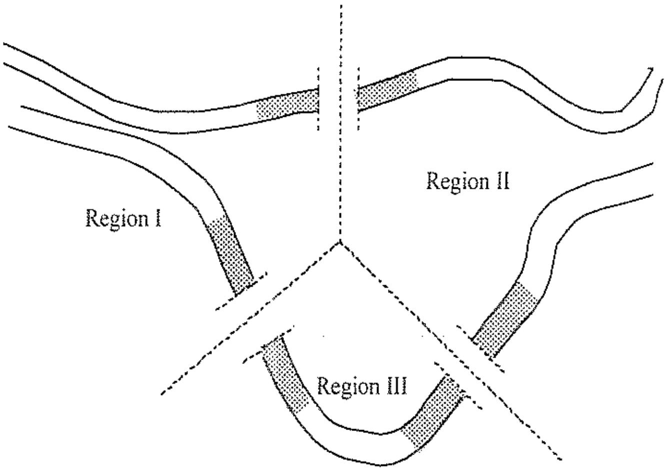

Figure 1. Partition of road network

(b) the acquisition of information from the other distributed processor.

3.2.1 Vehicle Transfer between Subareas

The system to be simulated is partitioned into N disjoint subsystems, and vehicles are free to move all over the road network. When a vehicle moves from one subarea to another (Fig. 2), the simulation clocks in both subareas must be synchronized. If the simulation clocks are not synchronized, the simulation system may transfer the vehicle in the future or in the past. The simulation would thus not be carried out correctly, nor would it correspond exactly to the real world. It is necessary to synchronize the simulation clocks among the processors.

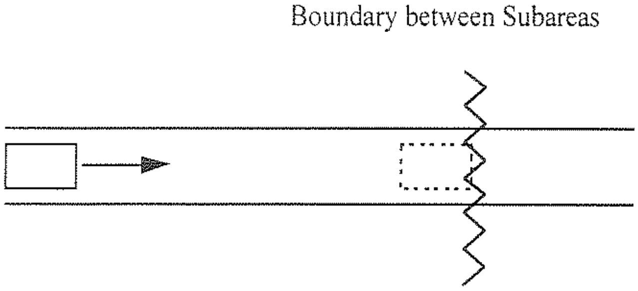

Figure 2. Vehicle-transfer between Subareas

3.2.2 Acquisition of Information

The second component of overhead time (b) is the time needed to obtain from other subareas the information necessary to process an event of the vehicle (Fig. 3). This information may be needed in almost all parallel simulations.

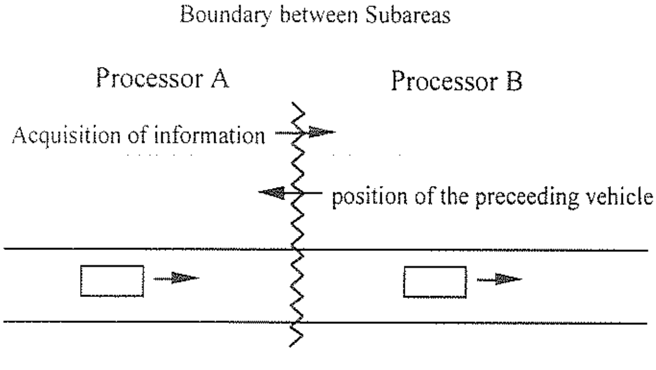

Figure 3. Acquisition of Information

In our simulation, the decision-making for driving depends upon various factors such as velocity, degree of congestion, and so on. It is also generally agreed that the distance between two vehicles is a very important factor in the model for vehicles in a road traffic simulation. We modified our model to take into consideration the distance between two vehicles. Accordingly, the decision-making of our vehicle model is strongly influenced by the position and velocity of the preceding vehicles. Our vehicle's model therefore always has to obtain the position and velocity of the preceding vehicle. The vehicle's model cannot obtain this information when the preceding vehicle has passed the next subarea. In this case, it is impossible to drive the vehicle.

4. NEW ALGORITHM

The overhead time for processing the synchronization of the simulation clocks among N distributed processors and for acquiring information from the other processors must be reduced.

4.1 Synchronization of the Simulation Clock

We use the CTW (Conservative Time Window) algorithm. This algorithm had previously only been used in discrete event parallel simulations. We now apply it to

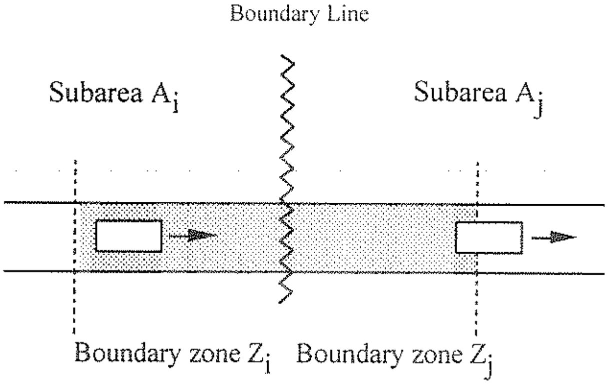

Figure 4. Boundary Line and Zone

a continuous event parallel simulation for a network-type system.

Our new method to synchronize the simulation clocks introduces the concept of boundary zones in road networks. The road is divided into two subareas, A_i and A_j, by a boundary line (Fig. 4). The concept of the subarea is equal to the concept of the region described before. Two boundary zones, Z_i and Z_j, are defined as both bordering sides of the boundary line. If a vehicle enters boundary zone Z_j from boundary zone Z_i, it is necessary to terminate the vehicle from the event chain of the processor for subarea A_i and generate the vehicle in the event chain of the processor for subarea A_j. Both the termination and generation of a vehicle are regarded as a discrete event. A vehicle should leave subarea A_i and enter subarea A_j in the same simulation clock. By introducing the concept of boundary zone, it is possible to process the synchronization smoothly among subareas.

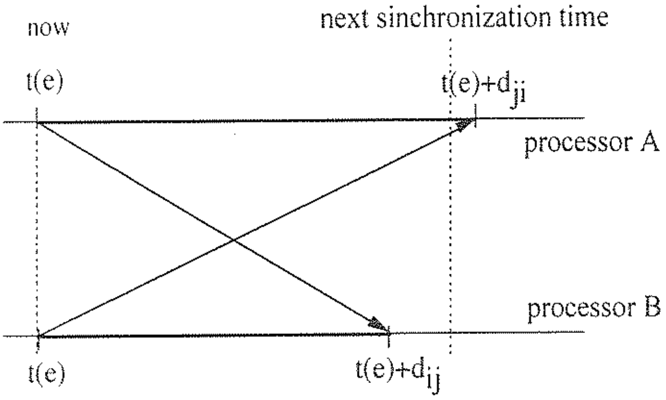

Figure 5. Forecast the next Sinchronization time

The new method we present here is based on forecasting the vehicle's arrival time in the next subarea A_j (Fig. 5). This arrival time is forecasted when a vehicle enters boundary zone Z_i. It is impossible to forecast the exact entrance time because a vehicle moves irregularly, depending on various factors.

So we estimate the minimum forecast time $\widetilde{d}_{ij}$ such that

$$\widetilde{d}_{ij} \leqq d_{ij}$$

For the minimum forecast time $\widetilde{S}_j$, the following equation holds:

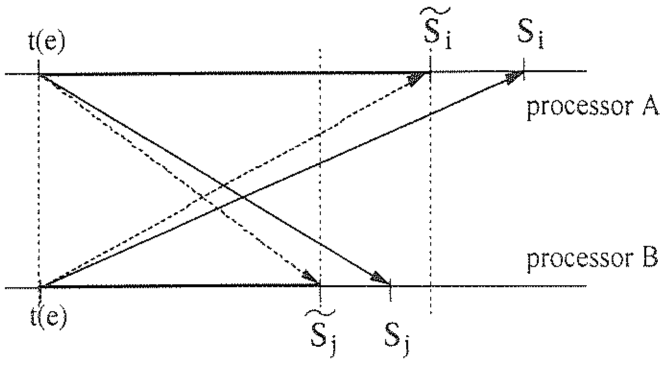

Figure 6. Forecast of the Event

$$t(e) + \widetilde{d}_{ij} = \widetilde{S}_j$$

We can thus obtain the relation (Fig. 6).

$$\widetilde{S}_j \leqq S_j$$

On the other hand, the system has to synchronize at time S_j. Thus, in almost cases the following relation holds:

$$\widetilde{S}_j \neq S_j$$

There arises another problem: how to achieve synchronization during the time from $\widetilde{S}_j$ till S_j.

This system continues making time window till time S_j. That is,

$$\widetilde{S}_{j(1)} < \widetilde{S}_{j(2)} < \cdots < \widetilde{S}_{j(m)} \approx S_j$$

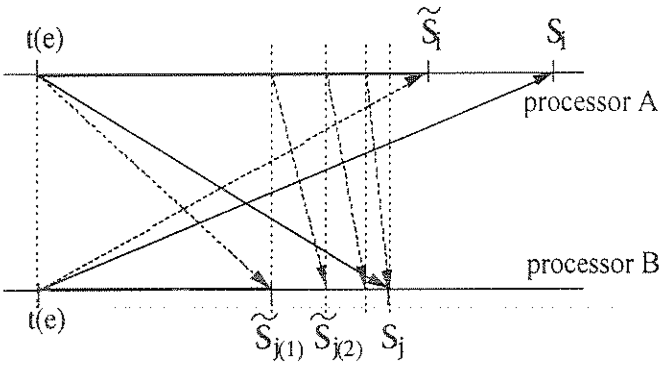

Figure 7. Forecast of Vehicle-Transfer Time

This method repeats forecasting the vehicle's entrance time to the next subarea, as shown in Figure 7.

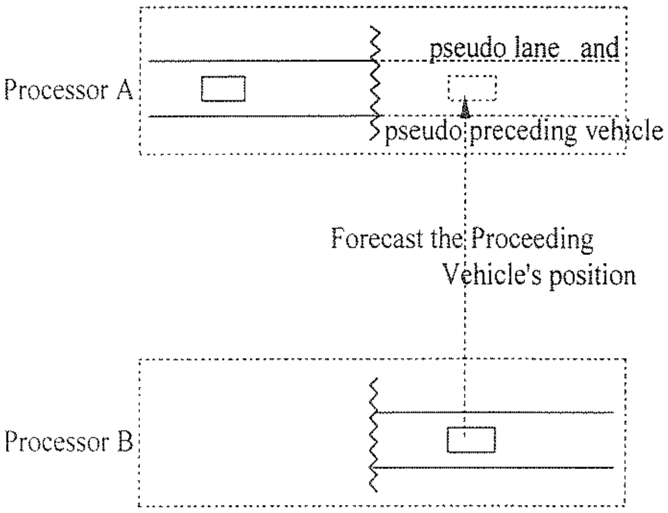

Figure 8. Setting of Pseudo Lane

4.2 Setting of Pseudo Lane

The vehicle's model cannot obtain the information when the preceding vehicle passes into the next subarea. In this case, it is impossible to drive the vehicle. To solve this problem we introduce the boundary zone concept described in Sec. 4.1. If the preceding vehicle is in the next boundary zone Z_j, the vehicle's model has to get the information on the preceding vehicle in the next boundary zone Z_j. We consider the following effective method for obtaining this information .

This method creates a pseudo lane, and a pseudo preceding vehicle runs along this lane. This method is established by forming a pseudo lane in boundary zone Z_i as if another lane different from the actual lane were in next boundary zone Z_j connected to boundary zone Z_i, and a pseudo vehicle ran along this pseudo lane in subarea A_i. As a result of this pseudo lane and vehicle, two vehicles run along an individual lane, as shown in Figure 8.

5. VERIFICATION OF OUR ALGORITHM

We have used our simulation system, which consists of two personal computers connected by a communication cable, to carry out two kinds of simulations as following (2), (3).

(1) Non Parallel simulation: A road traffic system is carried out by a single processor.

(2) Full synchronization method: Parallel simulation is carried out by synchronizing every simulation clock in each processor at every simulation clock.

(3) The method using the new algorithm on parallel simulation: The synchronization of the simulation clocks is carried out by the method of forecasting the arrival time in the next subarea. If the forecasted and actual arrival time are not the same, the arrival time of the vehicle to the next subarea is forecasted again. This procedure is continued until the forecasted time and the actual time become equal.

This simulation is carried out with a model that generates a pseudo lane.

With this system, we have carried out parallel simulation for the road traffic system.

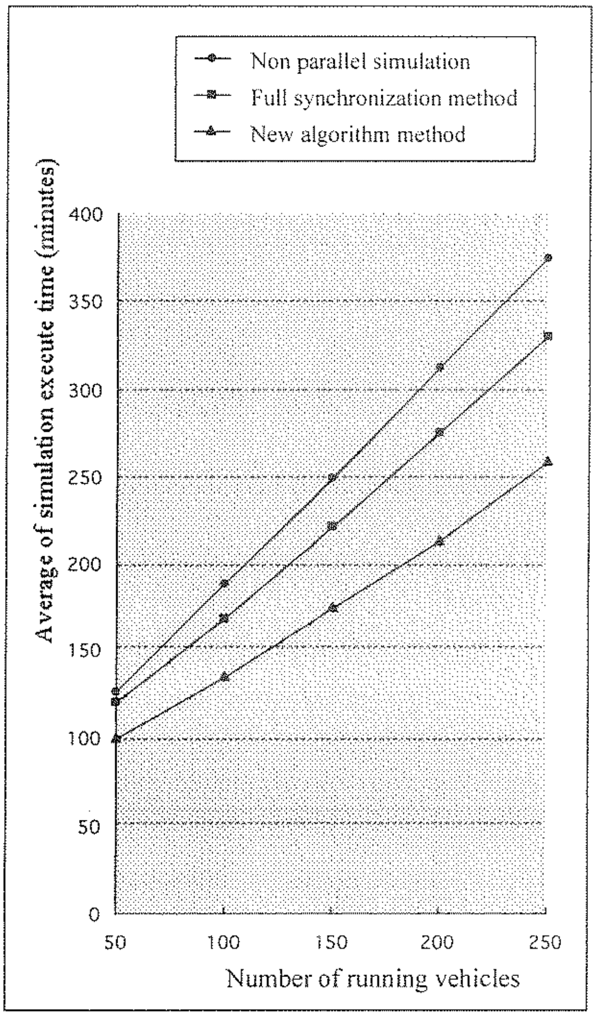

Figure 9. Results of the Simulation

6. CONCLUSION

We have proposed a new parallel simulation algorithm that extends the conservative approach, and we have verified the algorithm's effectiveness. We achieved this by applying Conservative Time Window method for discrete-change model to continuous- and discrete-change model. More in-depth study is needed to refine this algorithm and reduce the gap between the forecasted event time and the actual one in the next subarea, and to carry out parallel simulation using more than two processors. Moreover, an issue to be address in the future is to apply other network systems.

7. ACKNOWLEDGMENTS

This study is supported the members of project MITRAM (road traffic simulation system using microscopic model for analyzing traffic jams). We thank them for their contributions.

REFERENCE

Atsuo OZAKI, Ryozo KIYOHARA, et al.,PARALLEL SIMULATION BASED ON SPACE-TIME OBJECT MODEL, The 13th simulation technology conference in JAPAN, Tokyo, pp299-302, 1994

David M.NICOL, Scott E. RIFFE, A "CONSERVATIVE" APPROACH TO PARALLELIZING THE SHARKS WORLD SIMULATION, Proc. of 1990 WSC

Fujimoto, R.M. PARALLEL DISCRETE EVENT SIMULATION. Communications of ACM. October1990

Gang-Len CHANG, Thanavat JUNCHAYA, SIMULATING NETWORK TRAFFIC FLOWS WITH A MASSIVELY PARALLEL COMPUTING ARCHITECTURE, Proc. of 1993 WSC, pp762-770

Mani CHANDY, Jayadev MISRA, DISTRIBUTED SIMULATION: A CASE STUDY IN DESIGN AND VERIFICATION OF DISTRIBUTED PROGRAMS, IEEE trans. of software eng. Vol se-5, No. 5, pp441-452

Michael PIDD, OBJECT ORIENTATION & THREEPHASE SIMULATION, Proc. of 1992 WSC, pp689-693

Rajive L. BATRODIA, Wen-Toh LIAO, PARALLEL SIMULATION OF THE SHARKS WORLD PROBLEM, Proc. of 1990 WSC

Rassul AYANI, Hassan RAJAEI, PARALLEL SIMULATION USING CONSERVATIVE TIME WINDOWS, Proc. of 1992 WSC, pp709-717

Toshio NAKANISHI, et al.,THE DEVELOPMENT OF A ROAD TRAFFIC SIMULATION SYSTEM IN BROAD AREAS, Proc. of MODSIM, Austoralia, 1993