Synchronization method for continuous-event parallel simulation

Mitsuhiro NAMEKAWA [a], **Hideki MORI** [b], **Akira SATOH** [b] **and Toshio NAKANISHI** [c]

[a]College of Management and Economics, Kaetsu University, Tokyo, Japan.

[b] Faculty of Engineering, Toyo University, Saitama, Japan.

[c] Faculty of Policy Studies, Tokushima Bunri University, Tokushima, Japan.

Abstract: We propose a new synchronization method for continuous-event parallel simulation in the transaction flow-type system, in which a certain kind of transaction moves in the three-dimensional space and constitutes a social system, such as road traffic systems, telecommunication systems, transport systems, and so on. The parallel simulation technique has attracted researchers' attention recently as a method that enables one to perform simulations that had previously been considered impossible in large-scale and complex systems, while improving the processing efficiency of the simulation and shortening the execution time.

Parallel simulation extracts the parallel property at the time of executing the simulation, and improves the simulation-processing efficiency by distributing the workload among multiple processors. However, almost all conventional parallel simulation is based on discrete events, while continuous-event parallel simulation has been but rarely used. This is because parallel simulation for continuous events requires, in principle, the synchronization of processing per simulation clock, so that the processing efficiency due to the parallel processing implemented deteriorates to a greater degree.

We propose a new synchronization method that achieves a desirably efficient simulation, because a certain synchronization is omitted by means of transport prediction of the transaction in the transaction flow type system.

In this paper, we describe the synchronization method to improve the processing efficiency of each processor. We have also carried out experimentation thereof, and present its efficiency, verification and validity by comparing our method with the synchronization way per simulation clock.

Keywords: *parallel simulation, transaction flow-type system, continuous-changed model, road-traffic system*

1. INTRODUCTION

The parallel simulation technique has recently become an effective method which enables us to perform simulations in systems which, because of their large size and complexity have been considered impossible, while improving the processing efficiency of the simulation and shortening the execution time.

Parallel simulation extracts the parallel property at the time of executing the simulation and improves simulation-processing efficiency by distributing the workload among multiple processors. However, almost all conventional parallel simulation is based on discrete events, and continuous-event parallel simulation is rarely used. This is because parallel simulation for the continuous-event requires, in principle, synchronization of processing per simulation clock, with the results that processing efficiency deteriorates. We classify simulation modes into two systems. One is a spatial flow-type system expressed by a model which reproduces the situation where fluid flows in three-dimensional space. The other is a transaction flow type system, in which a certain kind of transaction moves in the space.

On the other hand, there are two methods for distributing the workload among multiple processors in a parallel simulation. In one method, a specific function in common is made separate and independent so as to parallel-process the function. In the other method, a system as an object for the simulation is divided spatially, so that the process is divided for each divided sub-space. The former is defined as the function-distributed method while the latter is defined as the space-distributed method.

In this paper, we propose a new synchronization method for continuous parallel simulation using the space-distributed method. This method achieves a desirably efficient simulation because certain synchronizations are omitted by means of transfer forecast of the transaction in the transaction flow-type system. It can simulate almost all social systems. Then, we will justify our method and verify its effectiveness.

We will describe this method of parallel simulation, especially by the synchronization processing per simulation clock in section 2, a new synchronization processing method in section 3, verification for the new method in section 4, present our evaluation in section 5 and finally our concluding remarks in section 6.

2. METHODS IN PARALLEL SIMULATION

In parallel simulation, a process is needed which avoids logical conflicts and which correctly holds the order of simulation clock between processors where the workload is distributed among them. As a result, its overhead time becomes problematic in terms of process efficiency.

2.1. Overhead time

The overhead time in the space distributed method depends on neighboring sub-space as well as on the transfer of transactions. It is to be noted that as long as the processors have the identical function and performance in parallel processing, the processing time for simulation alone does not change irrespective of what distributing mode is adopted. Therefore, in parallel simulation using the space distribution method, it is necessary to add preparation time, communication time, waiting time and so on for consistency processing as well as synchronization processing, to the simulation time.

2.2. Synchronization processing

Parallel simulation for the continuous event requires, in principle, synchronization processing per simulation clock. This method is simple because it does not need synchronization processing among processors. But processing efficiency due to the parallel processing implemented in computers system is deteriorated to a greater degree. On the other hand, we propose a method to improve the processing efficiency of each processor by omitting a part of synchronization processing from usual parallelism among processors. We abbreviate the former as the SMP method and the latter as the NSM method respectively. Timing methods for each are shown as Fig.1 and Fig.2 respectively.

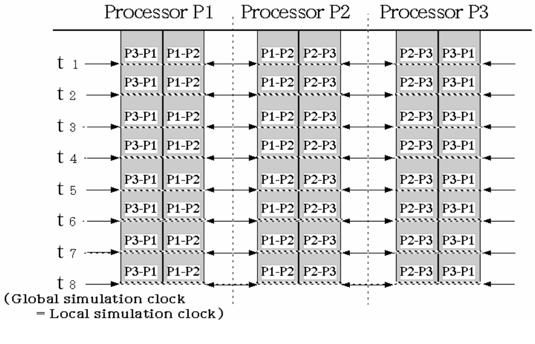

Fig.1 Timing of the SMP method

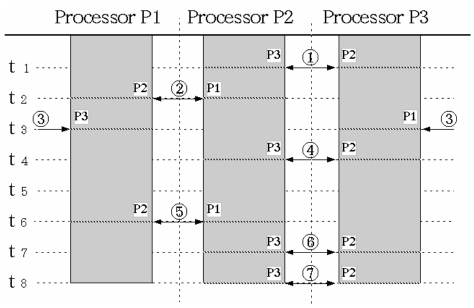

Fig.2 Timing of the NSM method

Synchronization processing is divided into three phases: preparation processing, communication processing among processors, and waiting processing for synchronization.

2.3. Effectiveness of parallel simulation

The number of synchronization times for the SMP method is the same as the number of simulation clocks. If it is possible to omit the time of synchronization processing by using effective technology, we are able to achieve a shorter simulation execution time (NSM(n)), as shown as Fig.3. SMP(n) is the execution time of parallel simulation with the SMP method using n processors. S means simulation execution time using one processor. S/n shows the average simulation execution time per processor. Fig.3 shows the relation ($S/n < NSM(n) \leqq SMP(n)$) among three types of synchronization processing methods: S (single processor), the NSM, and the SMP, depending on n processors. The goal of our study is to obtain a more effective synchronization processing method (NSM).

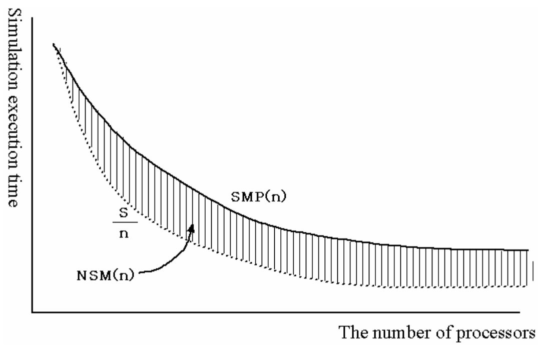

Fig.3 Effectiveness of parallel simulation

3. NEW SYNCHRONIZATION PROCESSING METHOD

In this section, a new synchronization processing method for the parallel simulation will be described.

3.1. Discrete event model building

Simulation, in which vehicles are expressed as microscopic models such as a road traffic system simulation, is considered to be a typical continuous-event simulation. A characteristic of the continuous-event simulation is that the change of the system is mainly dependent on continuous change (that is, continuous movement of vehicles in the road traffic system). Thus, the movement of a vehicle (transaction) in the road traffic system is simulated in a manner which is described as a continuous-change model. The road traffic system is considered to be representative of the space distributed method. In the parallel simulation implementing this distribution method, the space to be simulated is partitioned into disjointed sub-spaces. A processor is assigned for each divided sub-space, so that simulation corresponding to each sub-space can be carried out. Thus, a vehicle in the road traffic system which moves through the simulation area (space) moves from one sub-area (sub-space) to another, that is, from one processor to another. We define this movement as 'transfer'. However, it is to be noted that these transfers do not necessarily occur per simulation clock, though their occurrence depends on the density of traffic. In other words, it is found that the transfer of transactions between sub-spaces is a discrete event.

As for the continuous-event model, parallel simulation synchronization itself is expressed as a discrete-like event. We are trying to create a parallel simulation method by which the transfer of a transaction between processors gives out the same result as if it had been obtained by a simulation performed by a single processor.

3.2. Adoption of the transfer forecast concept

In order that the transfer of a transaction between sub-spaces be performed consistently with the same timing, it is indispensable that the time of the transfer be accurately known in advance. However, since the transfer is generally not deterministic, it is impossible to carry out the transfer forecast in a smooth and efficient manner. Hence, we attempted to achieve a smooth and efficient transfer forecast as precisely as possible. The forecast which turns out to be earlier than the actual transfer time results in the increase of the execution time. However, since the forecast is not behind the actual transfer, a consistent synchronization process is guaranteed.

The transfer from one sub-space to another can be roughly predicted by taking the current location, transfer direction and transfer speed into consideration so that the arriving time and destined sub-space can be known. Since the transaction transfer changes generally in a random manner, it is necessary to carry out the transfer forecast based on each term, such as the current location, transfer direction, and transfer speed. In order that this transfer be carried out smoothly, the simulation clocks between two neighboring sub-spaces must be identical. Therefore, it is indispensable that the transfer forecast is ahead of the actual transfer. If the forecast is behind the actual transfer, this method is no longer valid. If the first forecast does not match with the actual one, however, several more predictions can lead to a smooth transfer as shown in Fig. 4. If the synchronization as depicted in Fig. 4 is carried out, the simulation can be performed independently at each sub-space alone, without other sub-spaces concerned, up to the

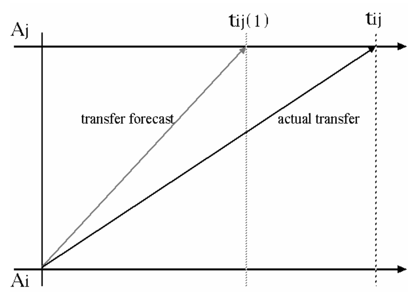

Fig.4 Transfer forecast of a transaction

synchronization time. As a result, the synchronization process at each simulation clock is omitted. Thus, the communication time is decreased and the total of simulation execution time is also reduced.

3.3. Decision of synchronization time

If sub-space l, m and n borders on sub-space j, and arriving time (at global simulation clock) of a transaction from three neighboring sub-spaces l, m and n to sub-space j is T(l,j), T(m,l), and T(n,l),respectively, then the synchronization time between sub-spaces n and j is defined by the following equation.

$$\min_{j}\{ T(l,j), T(m,j), T(n,j) \} = T(n,j) \qquad (1)$$

3.4. Effective communication method

It is necessary to communicate messages for synchronizing the simulation clock among processors. We adopt a multi-agent method to communicate such messages so as to cut down the time of communication. The system is divided into disjointed sub-spaces composed of agents and others (members). One sub-space (agent) borders several neighboring sub-spaces (members). We call this a cluster. And example of a two-dimensional space is shown in Fig.6. If we define the communication method for synchronization processing between an agent and members to communicate among processors for each cluster, it is possible to decentralize communication processing among processors and to communicate messages more effectively.

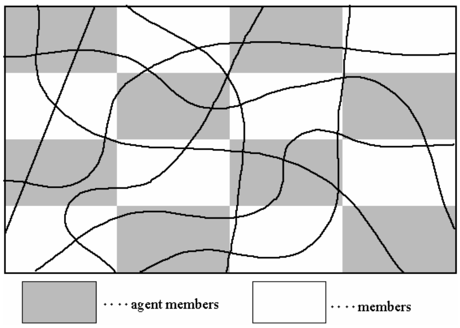

Fig.5 Classification of space

4. VERIFICATION FOR THE NSM METHOD

In order to verify our proposed algorithm so as to evaluate it, a simulation system has been developed. Several simulations for confirming and evaluating the effectiveness of the algorithm for its parallelism have been carried out. We call this evaluation simulation. The structural data on the overhead time in the parallel simulation and various data required for executing road-traffic simulation with the micro-model are obtained from the continuous-event parallel simulation.

4.1. Evaluation simulation

The evaluation simulation with multiple processors employed is performed virtually on a single processor as shown as Fig.6.

In this simulation, the effects of parallelism can easily be evaluated by varying parameters. For example, the number of processors is specified by a parameter, so the distributed parallel environment corresponding to the number of processors is automatically formed and the simulation to be evaluated is executed.

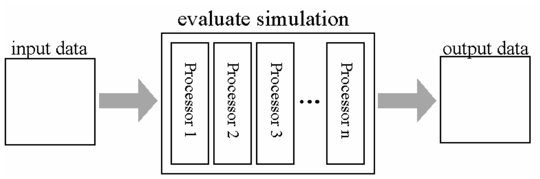

Fig.6 Function of evaluation simulation

(1) Input data

- (a) The number of processor.
- (b) The characteristics of the system to be simulated (road map (Fig.7)), the number of vehicles and places to be generated, maximum and average speeds of vehicles.
- (c) The characters of execution rime for parallel simulation on road traffic system (simulation execution time per each vehicle, preparation processing per each predicted transaction transfer, and communication processing time among processors, execution time per each synchronization processing)

(2) Flow of simulation

The evaluation simulation with multiple processors employed is performed virtually on a single processor. In this simulation, the effect of parallelism can easily be evaluated by varying the parameters mentioned above. The structural data on the overhead time in the parallel simulation and various data required for executing road traffic simulation with the microscopic model are obtained from the continuous-event parallel simulation using two workstations (i.e, two processors).

- (a) Simulation execution time per each vehicle to run: 10ms
- (b) Execution time of preparation processing per transfer forecast of transaction: 20ms
- (c) Communication processing time among processors per each synchronization processing: 180ms

(3) Output data

- (a) Primary output data: road traffic density and the number of transactions to be transferred for each sub-area (processor), the times of synchronization processing and communication processing times, transfer forecast of transaction, queuing for synchronization processing between processors, execution time for road traffic simulation per every processors

(b) Secondary output data (statistical data): total execution time, synchronization processing time and overhead time, and improvement rate for parallel simulation

4.2. Road traffic system model

Suppose there is provided a circular road whose circumference is 60 km, as shown in Fig. 7. Then, the road is uniformly divided into sections (sub-roads). The vehicle's running speed is set to 40 plus or minus 5 km/hour in a uniform manner as an attributed value for the vehicle.

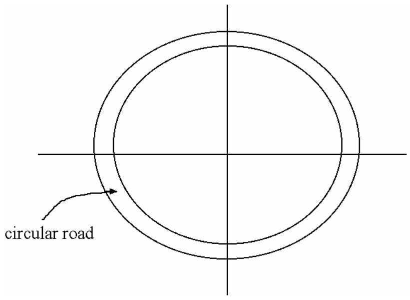

Fig.7 Road traffic system to be simulated

The vehicle is generated in a uniformly distributed manner and runs on the circular road. The vehicles are almost evenly distributed in this case. On the other hand, a scenario in which the vehicles to be processed contain twice as much as the not-evenly-distributed transactions in a specific section can be set up, too. The vehicles are not evenly distributed in this case.

4.3. Simulation cases

The parallel simulation is performed with a certain area being distributed. When the transactions are evenly distributed among each processor, the effect on the parallelism will be of a greater degree. However, in general, the distribution of the transaction is not well balanced since the movement of transaction occurs at random. Thus, in order to confirm the generality of the proposed algorithm, the simulation is performed on two kinds of cases: one is a case in which the transaction is almost evenly distributed; the other is a case in which the transaction is not evenly distributed.

The vehicle is generated in a uniformly distributed manner and runs on the circular road in the case where the transaction is almost evenly distributed. On the other hand, an environment in which the transaction to be processed contains twice as much as the not-evenly-distributed transaction in a specific area is set up in cases where the transaction is not evenly distributed.

4.4. Evaluation method

The rate of speedup improvement in executing the simulation is a general method of evaluating the parallelism effect. In our study, as a means to confirm the effect of the algorithm, the number of communications responsible for the overhead time as well as the waiting time for the synchronization in addition to the rate of speedup improvement are analyzed and evaluated.

5. EVALUATION

In this section we will verify the effectiveness of the new synchronization method (the NMS) by comparing it to the synchronization processing method per simulation clock (the SMP).

5.1. Effectiveness of the NSM method

Fig. 8 shows a result of the execution of the simulation where the workload is almost evenly distributed.

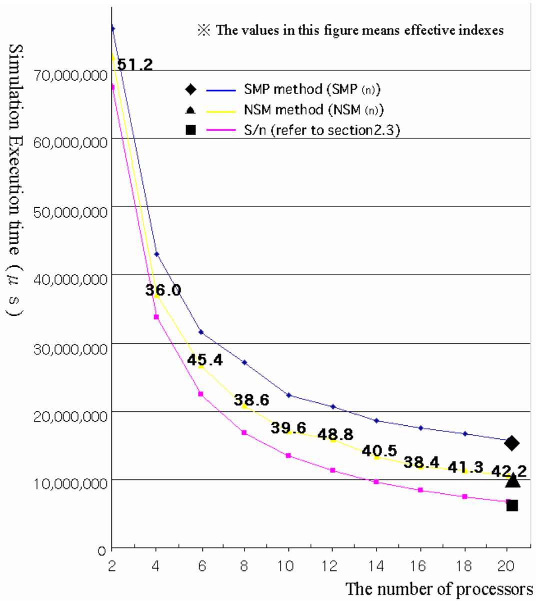

Fig.8 Simulation execution time and the indicate ratio

S/n of Fig.8 is not inclusive of the overhead time. Thus, this means pure simulation execution time per each processor without overhead time such as preparation, communication and queuing time mentioned in section 2.3.

Fig.8 shows the relation of simulation execution times by the NMP, the NSM, and S/n depending on n processors respectively. The numerical values of the graph in which is between the SMP(n) and S/n shows the effectiveness of the

NSM method. We call it effective index of a new synchronization processing method. It is defined by the following equation.

The effective index

$$= [NSM(n) - \{S/n\}]/[SMP(n) - \{S/n\}]*100 \quad (2)$$

This indicate ratio of the overhead time for the NSM method against the SMP method. Thus, the value of the effective index is larger, the new algorithm (the NSM) is better.

We find the efficiency of the NSM method from Fig.8. The average of effective indexes is 40%. We verified satisfactorily the effectiveness of the new synchronization method (the NSM).

5.2. Availability in the NSM method

We carry out several simulations changing the traffic density with 8 parallel processors to verify an availability of the NSM method. The result of the simulations is shown inFig.9. We can find out that both lines of the SMP method and the NSM one are crossing at high density (6,700 vehicles) from this figure. This indicates that the density in the road traffic system is beyond availability of the NSM method. In a word, the NSM method requires greater simulation execution time than the SMP method. This shows that the NSM method may be available within the density of about 5,000 vehicles.

6. CONCLUDING REMARKS

Increased quality has been guaranteed in the continuous-event parallel simulation, and a new synchronization processing method which enables

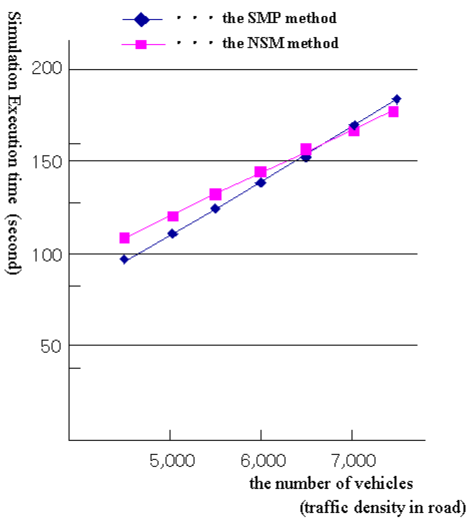

Fig. 9 Availability of the NSM method

us to perform simulations with improved processing efficiency has been incorporated into the congested road-traffic system. Then, experiments have been carried out in the continuous-event parallel simulation environment so as to verify the effects of the processing efficiency.

The synchronization method in this new continuous-event parallel simulation is a system of the transaction flow-type which can be found almost everywhere in the society in general. For example, the system includes a railroad transport system and information communication system.

In the future, we will examine possible applications to various other systems in the hope that we can analyze them in order to gain wider use and efficiency with even more efficient synchronization method.

7. REFERENCES

Mineo Takai, Tokuji Yamashiro, Seinosuke Narita: Evaluation of Conservative Synchronization Methods in Parallel Discrete Event Simulation. Vol.32, No.11, 1996.11

Hassan, R.,: Parallel Simulation Using Conservative Time Windows. Winter Simulation Conference, 709-717, 1992

Misra, 1986, Distributed Discrete-Event Simulation., ACM Comp., surveys. Vol18, pp.39-65.

M.Namekawa, A.Satoh, et al.: Clock Synchronization Algorithm for Parallel Road Traffic Simulation, International Congress on Modelling and Simulation, pp.1426-1431, 1997.12