Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, Volume 21, No. 1, ISSN 1726-9679
ISBN 978-3-901509-73-5, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2010
Make Harmony Between Technology and Nature, and Your Mind will Fly Free as a Bird
Annals of DAAAM International

IMPACT OF THE MESSAGE LENGTH ON ITS TRANSFER TIME IN DISTRIBUTED TRAFFIC SIMULATION

POTUZAK, T[omas]

Abstract: *In this paper, the dependency of the transfer time of a message on its length in the distributed simulation of road traffic is investigated. By a set of tests, it is verified that the transfer of lower number of longer messages is more efficient than the transfer of higher number of shorter messages. This feature is important for the efficiency of the communication reduction methods for distributed simulation of road traffic, which we developed. These methods significantly reduce the number of the messages sent in the simulation at the cost of the increase of the length of the transferred messages.*
Key words: *Messages, length, time, communication, simulation*

1. INTRODUCTION

The computer simulation of the road traffic is a very important tool for analysis and control of traffic networks. However, a detailed simulation of a large traffic network (e.g. an entire city) requires a great amount of computational power in order to be performed in a real or "better-than-real" time. Hence, many simulators of road traffic have been adapted for distributed computing environment, for example (Nagel & Rickert, 2001) and (Gonnet, 2001). In that case, each computer (node) of the distributed computer performs a process with simulation of one part of traffic network (a traffic sub-network).

The processes communicate among each other using messages. However, the inter-process communication is one of the main bottlenecks of each distributed application. Hence, in our research, we focus on development of an efficient communication protocol for distributed simulation of road traffic, which would minimize the message load of the distributed simulation.

So far, we developed several methods for reduction of inter-process communication, which are described in (Potuzak, 2008) and (Potuzak & Herout, 2009). These methods can be used for distributed road traffic simulation only. However, they can significantly reduce the total number of messages sent in the simulation at the cost of the increase of the messages length. If these methods shall be efficient, the increase of the messages length must have only negligible effect on the transfer time of the messages. This feature will be verified in this paper.

2. INTER-PROCESS COMMUNICATION

As it was said, in a distributed simulation of road traffic, each node of the distributed computer performs a process with simulation of one traffic sub-network.

2.1 Data Transferred in Distributed Traffic Simulation

The most important issue of the distributed traffic simulation is the communication protocol, which ensures the transfer of vehicles between the neighbouring traffic sub-networks and the synchronization of the simulation. The synchronization ensures that the particular simulation processes perform the same time step at the same moment. This is necessary in order that the vehicles sent from one traffic sub-network to another arrive in correct time step (Potuzak & Herout, 2009).

So, there are two types of data transferred in the distributed simulation of road traffic – the messages with vehicles and the synchronization messages (Klefstad et al., 2005).

2.2 Methods for Reduction of Inter-process Communication

During our research, we developed several methods for the reduction of the inter-process communication. These methods significantly reduce the total number of messages sent in the simulation at the cost of the increase of the messages length. The methods will be briefly described in this section.

The *centralized vehicle transfer* is a method for the reduction of the communication necessary for the transfer of vehicles. Its main idea is to transfer the vehicles added to the synchronization messages using a central control process as a message router (Potuzak, 2008). This approach indeed reduces the number of sent messages, but also increases the length of the particular synchronization messages.

Another approach, which we developed, is the *long step method*. Its main idea is to send the vehicles once every several time steps. The time period between two successive transfers of vehicles is designated as a *long step*. Using this approach, all vehicles, which shall be transferred to a neighbouring traffic sub-network throughout the long step, are stored in the buffer. After the long step period is elapsed, the content of the buffer is packed into a message and sent to the corresponding sub-network where the contained vehicles are forwarded to the target traffic lanes. Since the synchronization is necessary only because of the transfer of the vehicles between the sub-networks, the synchronization is also performed once per long step during the vehicle transfer (Potuzak, 2009). Similar to the centralized vehicle transfer, the long step method reduces the sent messages count, but increases the length of the messages.

3. MIDDLEWARE FOR MESSAGE TRANSFER

Besides the number of messages, the middleware is also important for the resulting performance of the distributed simulation. The middleware is a layer that provides services for inter-process communication (Klefstad et al., 2005). The particular middleware differ each other in speed, applicability, and the comfort of the programming. Because all the communication protocols we developed are implemented in Java, we are considering two applicable middleware – the TCP (Transmission Control Protocol) and the RMI (Remote Method Invocation). Both are briefly described in following sections.

3.1 TCP

The TCP is a protocol, which provides reliable communication. It creates connections and uses them for exchanging of data streams. It guarantees reliable transfer of data from sender to receiver. The data packets also arrive in order they were send (Grosso, 2002). Because the TCP supports only the transfer of raw data, there is no overhead caused by additional features such as object serialization. So, the main advantage of the TCP is the communication speed. The main disadvantage is that the encoding and decoding of transferred data must be implemented in the application level along with the business logic of the simulation. This can be error-prone.

3.2 RMI

The RMI is an object-oriented version of RPC (Remote Procedure Call) incorporated in the Java Core API. The RMI uses the object-oriented paradigm in the distributed environment. By its utilization, it is possible to invoke a method on an object that resides in different JVM (Java Virtual Machine). Moreover, the invocation of a remote method is very similar to the invocation of a local method. All common mechanisms known from local computing (e.g. propagation of exceptions, garbage collection) are provided (Grosso, 2002).

The RMI uses the TCP sockets for data transfer, but it acts as a local method invocation. Hence, the RMI must implement a mechanism for object serialization, distributed garbage collection, and distributed propagation of exceptions (Grosso, 2002). These additional features bring an overhead into the communication. So, the RMI is slower than the TCP sockets. On the other hand, no additional features (encoding and decoding of transferred data) must be implemented in the business logic of the simulation. So, the inter-process communication is quite comfortable and transparent.

4. TRANSFER OF LONG AND SHORT MESSAGES

As it was said, our methods for communication reduction in distributed road traffic simulation are based on the transfer of lower number of longer messages. If the methods shall be efficient, the transfer of lower number of longer messages must be faster than the transfer of higher number of shorter messages.

This assumption is based on the fact that a little overhead is associated with every message (Nagle, 1984). The messages do not contain useful data only, but additional information necessary for the functioning of the utilized middleware as well. The lower the number of message is, the smaller is also the total communication overhead (see Fig. 1). This overhead differs for various middleware. For example, the communication overhead associated with every message is significantly higher using the RMI than using the TCP, because the RMI ensures additional features such as distributed garbage collection (see Section 5).

5. TESTS AND RESULTS

In order to verify the assumption that the transfer of lower number of longer messages is more efficient (or faster) than the transfer of higher number of shorter messages, a set of tests was performed. The tests were performed on a cluster called *Hydra*, which is available at our department. Each node of the cluster includes one processor Intel Xeon 3.2 GHz, 2 GB RAM, and 80 GB HDD. The nodes are interconnected by 1 Gb Ethernet.

Because our research is focused on the distributed simulation of road traffic, we utilize the transfer of vehicles for the tests. There were from 4 to 20 vehicles to transfer between two processes. First, all the vehicles were sent in one message. Then, each vehicle was sent in a separate message. Both approaches were compared according to the transfer time of the message/messages. In order to make the transfer time measurable, the dispatching of the message/messages was repeated thousand times per measurement. The tests were performed using both the TCP and the RMI.

The results are summarized in Tab. 1 where can be observed that the transfer of the lower number of longer messages is faster than the transfer of the higher number of shorter messages for both the RMI and the TCP. However, the difference is much more significant for the RMI. This is caused by the higher communication overhead associated with every message using the RMI. For 20 vehicles in each long message, the difference between the transfer times reaches up to 88 % using the RMI and up to 54 % using the TCP. The last observation is that the TCP was on average 15 times faster than the RMI by the transfer of the longer messages and even 59 times faster by the transfer of the shorter messages.

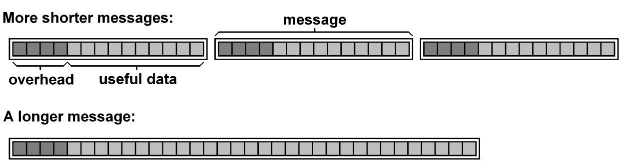

Fig. 1. Communication overhead of short and long messages

	RMI – time [ms]		TCP – time [ms]	
Vehicles count	Long message	Short messages	Long message	Short messages
4	468	1150	31	34
8	488	2539	32	42
12	498	2996	32	52
16	528	3921	32	62
20	541	4841	32	70

Tab. 1. Transfer times comparison of long and short message

6. CONCLUSION

In this paper, the dependency of the transfer time of a message on its length is investigated. The tests confirm that the transfer of the lower number of longer messages is more efficient than the transfer of the higher number of shorter messages (see Section 5). Hence, the methods for inter-process communication reduction, which reduce the number of messages at the cost of the increase of their length, can show significant savings of the inter-process communication time.

In our future work, we will focus on further methods for reduction of inter-process communication, based on reduction of both count and length of the messages.

7. ACKNOWLEDGEMENT

This work is supported by Ministry of Education, Youth, and Sport of Czech Republic - University spec. research - 1311.

8. REFERENCES

Gonnet, P.G. (2001). A Queue-Based Distributed Traffic Micro-Simulation, *Avaible from:* http://e-collection.ethbib.ethz.ch/eserv/eth:26374/eth-26374-01.pdf *Accessed:* 2010-06-11

Grosso, W. (2002). *Java RMI*, O'Reilly & Associates Inc., ISBN 1-56592-452-5, USA

Klefstad, R.; Zhang, Y.; Lai, M.; Jayakrishnan, R. & Lavanya, R. (2005). A Scalable, Synchronized, and Distributed Framework for Large-Scale Microscopic Traffic Simulation, *The 8th International IEEE Conference on Intelligent Transportation Systems*, pp. 813-818, Vienna

Nagel, K.; Rickert, M. (2001). Implementation of the TRANSIMS Micro-Simulation. *Parallel computing*, Vol. 27, No. 12, pp. 1611-1639

Nagle, J., 1984. Congestion Control in IP/TCP Internetworks. *ACM Computer Communication Review*, Vol. 14, No. 4, pp. 11-17

Potuzak, T. (2008). Centralized Transfer of Vehicles in Distributed Simulation of Road Traffic, *The 2nd Young Researchers Conference on Applied Sciences - Conference Proceedings Book*, pp. 42-47, ISBN 978-80-7043-698-1, Pilsen

Potuzak, T., Herout, P. (2009). An Efficient Communication Protocol for Distributed Traffic Simulation: Introduction of the Long Step Method, *Sofsem 2009: Theory and Practice of Computer Science, Proceedings, Volume II*, pp. 72-83, ISBN 978-80-7378-059-3, Špindlerův Mlýn