UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

A Methodology for Performance Evaluation of Optimistic Distributed Simulation Synchronisation Mechanisms

Renata Spolon
Marcos José Santana
Regina Helena Carlucci Santana

Nº 53

NOTAS

São Carlos - SP

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

ISSN 0103-2577

A Methodology for Performance Evaluation of Optimistic Distributed Simulation Synchronisation Mechanisms

Renata Spolon
Marcos José Santana
Regina Helena Carlucci Santana

Nº 53

NOTAS

Série Computação

São Carlos – SP
Dez./2000

A Methodology For Performance Evaluation Of Optimistic Distributed Simulation Synchronisation Mechanisms

Renata Spolon[1], Marcos José Santana[2], and Regina Helena Carlucci Santana[2]

[1] Departamento de Computação e Estatística – Centro de Ciências Exatas e Tecnologia – Universidade Federal de Mato Grosso do Sul
Cidade Universitária, S/N – C. P. 549 79070-900 Campo Grande, MS - BRASIL
renata@dct.ufms.br

[2] Departamento de Computação e Estatística – ICMC – Universidade de São Paulo
Av. Dr. Carlos Botelho 1465, C. P. 668 – 13560-970 - São Carlos, SP - BRAZIL
(mjs,rcs)@icmc.sc.usp.br

Resumo. Este trabalho apresenta um novo modelo para representar o protocolo de sincronização Time Warp, com o objetivo de avaliar seu desempenho em comparação aos protocolos variantes que vêm sendo apresentados na literatura. O procedimento de avaliação utiliza o modelo proposto como um núcleo onde as características dos protocolos variantes do Time Warp podem ser facilmente inseridas por meio de "plug-ins". O ambiente ALPHA/Sim é utilizado para a modelagem e simulação através de Redes de Petri. Desta forma, tanto usuários com experiência em simulação distribuída quanto aqueles que não dispõem de conhecimento profundo na área podem utilizar a metodologia para escolher o mecanismo de sincronização mais adequado.

A Methodology For Performance Evaluation Of Optimistic Distributed Simulation Synchronisation Mechanisms[1]

Renata Spolon[1], Marcos José Santana[2], and Regina Helena Carlucci Santana[2]

[1] Departamento de Computação e Estatística – CCET – Universidade Federal de Mato Grosso do Sul
Cidade Universitária, S/N – C. P. 549 79070-900 Campo Grande, MS - BRASIL
{renata@dct.ufms.br

[2] Departamento de Computação e Estatística – ICMC – Universidade de São Paulo
Av. Dr. Carlos Botelho 1465, C. P. 668 – 13560-970 - São Carlos, SP - BRAZIL
(mjs,rcs)@icmc.sc.usp.br

Abstract. This paper presents a novel model to represent the Time Warp synchronisation protocol aiming at evaluating its performance when both the basic protocol and several variations proposed over the last years are considered. The model proposed is the kernel of an evaluation procedure that considers a basic model (representing the original Time Warp mechanism) in which the variations can be easily introduced by means of plug-in to test different options for the synchronisation of distributed simulation programs. The model is built by means of Petri Nets, simulated using the ALPHA/Sim environment. The model and the overall methodology proposed are exercised through a basic example from the computer science field. The preliminary results show that the model and the methodology can be successfully used by both experienced and inexperienced users in order to choose the more appropriated synchronisation mechanism.

1 Introduction

Distributed simulation has experienced a significant development over the last twenty years, having become a powerful tool for performance analysis of complex systems. The major problem with distributed simulation is the necessity of synchronisation between processes, solved by a large number of variations in the mechanisms proposed in the literature. These mechanisms are basically classified as conservatives and optimistics. Conservative approaches, such as the CMB protocol, solve causality errors by avoiding event execution unless it is a safe operation [5]. Optimistic methods solve the dependence between simulation events by allowing violations to occur

[1] Published in Proceedings of The 14th European Simulation Multiconference 2000 (ESM'2000), Gent, Belgium, May 23-26, 2000, 121-125, ISBN 1-56555-204-0.

and then solving the problem using a rollback mechanism; the major example is Time Warp [2, 3]. One advantage in using distributed simulation is the reduction in the simulation execution time realised. However, the over-optimistic behaviour of Time Warp can degrade the performance of this protocol, and it has motivated many alternatives to bound this excess [1].

Furthermore, the choice among conservative and optimistic mechanisms is really a hard task that needs to consider not only the problem under study but also the hardware and software platforms, as well as the individual mechanisms characteristics. In this way, the several hybrid approaches that have been proposed in the literature try to insert some optimism into conservatives mechanisms or to limit the excess of optimism in the optimistic mechanisms. The work described in this paper deals with mechanisms that try to control the Time Warp optimism. These mechanisms are grouped according to their basic features into windows-based, probability-based, penalty-based, etc. [7]. This large variety of synchronisation mechanisms makes difficult the choice when someone is using distributed simulation and also makes comparisons hard. This fact emphasises the need for a methodology that allows the user to choose the mechanism that is the best to the requirements of the study, without the obligation of constructing it in advance. This paper presents the basic model and the set of parameters used to analyse the performance of Time-Warp-based synchronisation mechanism. The next sections present the behaviour of a Time Warp Logical Process (LP) needed to define the model and the associated performance evaluation methodology. The model is built by mean of Petri Nets [4] and this choice is adequate because Petri Nets can easily represent concurrence and show the details of each process activity.

2 Methodology for Performance Evaluation

Performance evaluation of a synchronisation mechanism or a synchronisation protocol can be made using the same tools applied for computing systems performance evaluation: Data Collection, Benchmarks and Prototypes (Measure Techniques) or Model Solution through Analytical Analysis and Simulation (Modelling Techniques). The use of such techniques for analysis of distributed simulation synchronisation mechanisms has the same advantages and disadvantages that when used for computing systems, but the most part of the published studies with Time Warp and its variants use specific test cases [7].

One way to analyse the performance of a protocol, without implementing it, consists of using modelling techniques to describe a model that represents the protocol behaviour and then giving parameters and exercising this model. Simulation comprises a suitable tool when it is impossible to measure directly into the system (in this research work there is not an implementation of the synchronisation mechanism). Considering that the actual implementation of all the synchronisation mechanisms to make the analysis is not an attractive approach, this research work uses modelling techniques and simulation to construct a methodology that permits simple and fast analysis of Time-Warp-based synchronisation mechanisms.

The performance evaluation was built form the establishment of a Petri Net model to represent the behaviour of an LP using the Time Warp protocol. By using this model the performance of a set of LPs can be simulated. The user needs only to give the suitable parameters and gets an estimate for the behaviour of the distributed simulation as a function of the features of the synchronisation mechanism, of the application and of the platform where the distributed simulation is going to be executed.

Small modifications, by means of plug-in, on the Time Warp basic model permit to represent its variants such as the windows-based mechanisms or the probability-based mechanisms. Thus, the use of a plug-in for each variation of the Time Warp mechanism makes easy the choice of the appropriated mechanism.

A more experienced user, expert in Petri Nets, can easily interact with the simulation environment and introduce his own plug-in sub-models. On the other hand, when either the knowledge of the user is not enough or he is looking for an easy way to build his performance evaluation model, the several previously built plug-in available in the environment can be used to evaluate the performance of different Time Warp variants.

The Time Warp protocol can be viewed as a collection of Logical Processes exchanging information about the required synchronisation to realise the distributed simulation by means of messages and antimessages. This behaviour is shown in Figure 1 that shows a block diagram for the behaviour of a logical process using the Time Warp to synchronise the distributed simulation. Basically, a Time Warp logical process deals with reception and execution of messages and antimessages that in turn can cause new messages and antimessages to be sent.

The reception of messages and antimessages (both sent by remote LPs) arriving from the communication network is represented by the Receiving block. The following actions are taken when receiving a message:

- The first action is to verify if an antimessage corresponding to the message occurred. In this case the antimessage is removed from the input queue and both are discarded (Cancel Ant block).
- If no antimessage corresponding to the message occurred, the LP verifies the message timestamp comparing its value to the Local Virtual Time (LVT):
- If the timestamp is greater than or equal to the LVT the message is stored in the input queue and will be executed latter (Store Msg block).
- If the timestamp is lower than the LVT, then a causality error occurred and the LP suffers a rollback returning the LP to a consistent state. In this case the antimessages can be sent to correct possible erroneous computation (Rollback block).

If the LP receives an antimessage, the following actions will take place:

- If the antimessage arrived before the corresponding message, this is stored in the antimessage queue (Store AntMsg block).
- If the message corresponding to the antimessage is already at the LP and it has not been executed, an antimessage kills it and both are discarded (Cancel Msg block).

- If the corresponding message arrived and was already executed, the LP must execute a rollback to return the simulation to a safe state, possibly involving the issuing of new antimessages (Rollback block).

Every execution of a message in the Time Warp involves three actions related to the creation of new events (Local Running block). An LP always removes a message with the lowest timestamp form the Input Queue and executes the corresponding event, the following actions can be considered:

- The generation of a new event that must be executed in a remote LP and then a message must be sent to the remote LP (Send Msg block).
- The generation of a new event to be executed locally and in this case a new message is punched into the Input Queue for latter execution (Store New block).
- It is also possible that the event executed does not produce any other event (Next Msg block) and in this case the LP fetches the next message in the Input Queue to execute.

Considering the basic block diagram shown in Figure 1, the points of interest comprise the manipulation of the messages and antimessages. An important task in the Time Warp is the process state-saving aiming at allowing the rollback mechanism to return the LP to a safe state, when causality errors occur. Several algorithms have been proposed to do it but they are not approached in this paper.

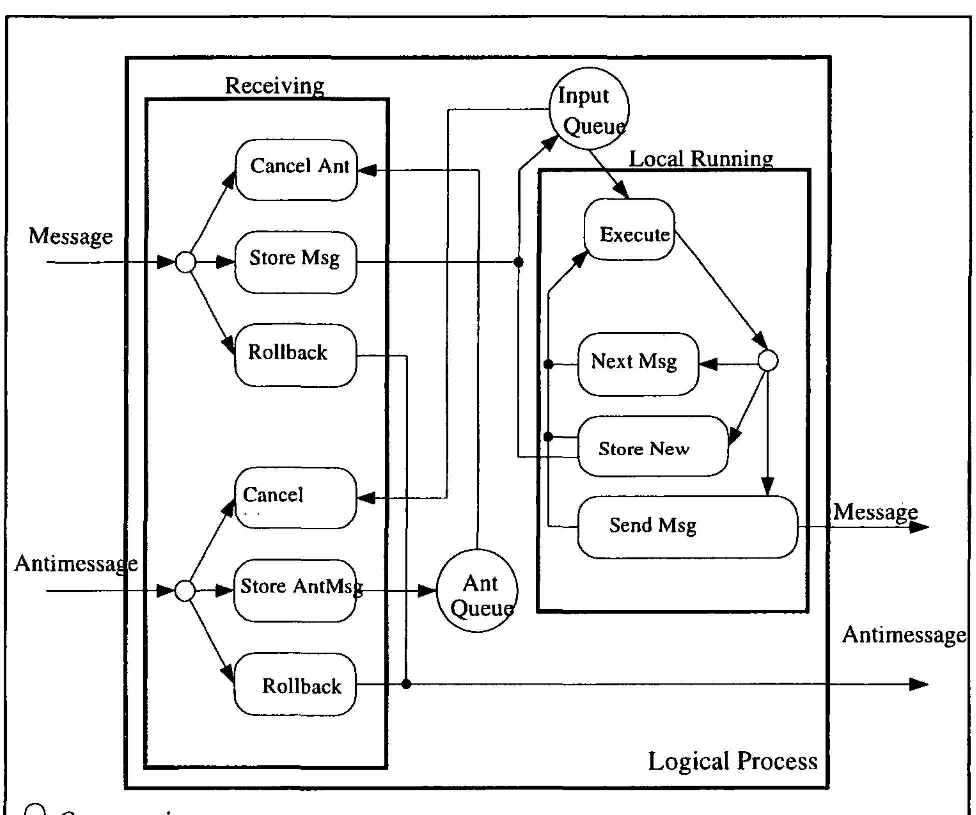

Fig. 1. Logical process abstraction.

The proposed performance evaluation methodology considers each Time Warp LP as a block (as described in Figure 1) that sends and receives messages and antimessages. From the users viewpoint the communication among the several LPs is the important point, rather than the way the synchronisation occurs inside each LP. Aiming at simplification, only an LP is considered executing in each processing element. The LPs are interconnected through a communication network as shown in Figure 2.

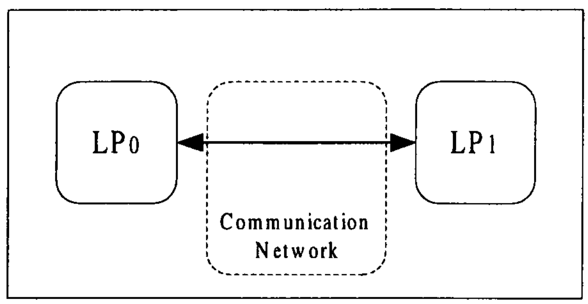

Fig. 2. LPs connected by a communication network.

Describing only the behaviour of a Time Warp LP is not enough for the performance evaluation purposes. The construction of a model representing completely the operation of an LP requires the definition of suitable parameters. The next section shows an overview of the environment built to perform the performance evaluation.

2 Environment for Performance Evaluation

The set of parameters needed to instrument the model to allow the performance evaluation of the Time Warp mechanism includes parameters related to the synchronisation mechanism (i. e., the behaviour of the LPs), to the platform used to implement the distributed simulation and to the features of the application described in the distributed simulation.

Figure 3 presents an overview for the proposed environment showing the relation between the simulation program and the input parameters.

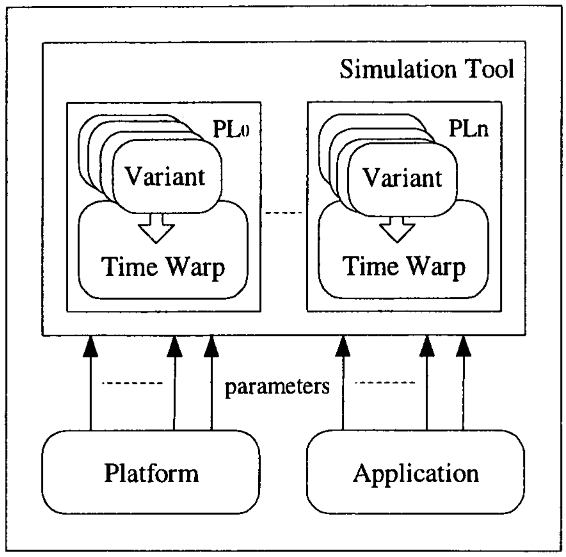

Fig. 3. Performance Evaluation Simulation Environment

The simulation tool is based on a Petri Net model for the Time Warp and its variants. The use of Petri Nets for complex systems performance evaluation depends on the availability of tools that allow the graphical model construction and its resolution. In this sense the ALPHA/Sim simulation tool [6] was selected because it allows the graphical construction of the simulation model, the simulation execution and the visualisation of results in a graphical environment. ALPHA/Sim uses the Generalised Stochastic Petri Nets and Coloured Petri Nets concepts to provide an easy to use, graphical interface, general-purpose discrete-event simulation tool that offers many statistics for performance evaluation [6].

The simulation program receives the information related to the application (the number of service centres and the way they are interconnect, the inter-arrival times, the service times, if it is an open model, etc.); the synchronisation mechanism (the Time Warp or one of its variations); the hardware platform (communication time, event execution time, time required to handle the data structures - the location of messages for local execution, state queue, output queue, input queue).

Figure 4 presents a basic block comprising an LP as seen in the performance evaluation methodology proposed. Basically, an LP sends and receives messages and antimessages to and from remote LPs, by means of the communication network. In this case the parameter required encapsulates the time needed to prepare, send and receive the message packet. If the simulation considers an open model then the inter-arrival time must be provide to the LP in order to start an arrival generator (λ) to feed the simulation program. In this case it is also required a sink element to represent the exit of clients. All these parameters are related to the application.

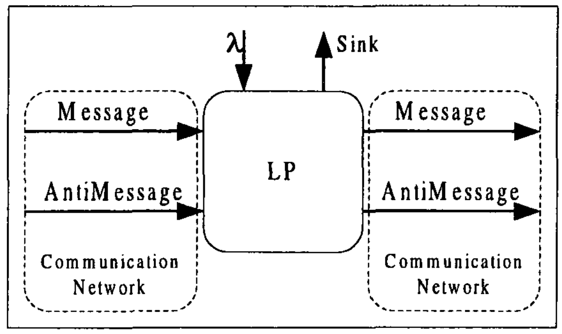

Fig. 4. Basic block.

3 Case Study

The queuing model showed in Figure 5 is the application chose to exemplify the use of the performance evaluation methodology described in this paper. This model describes a simplified computing system composed of a CPU and a disk unit. The jobs, arriving at the system, are serviced by the CPU and follow to the disk unit or leave the system with a probability p.

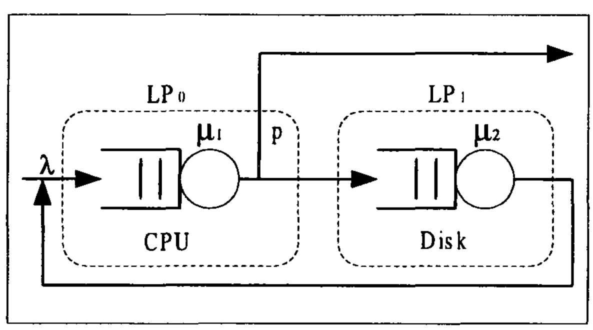

Fig. 5. Application.

The simulation of the Time Warp mechanism used in the application of the figure 5 considers two LPs. Each LP is simulated bye a Time Warp basic block as presented in figure 6. The LP_0 executes the messages corresponding to the CPU activities and LP_1 the disk ones. The parameters λ, μ_1 and μ_2 are used by the LPs to produce time-stamps for the new messages (λ, μ_1 for LP_0 and μ_2 for LP_1); p is the probability of a job leaving the system, implying that no messages will be issued from LP_0 to LP_1. As the disk, represented by LP_1, always receives and send requests from/to the CPU (LP_0), the LP_1 does not need to have the λ input nor the sink element.

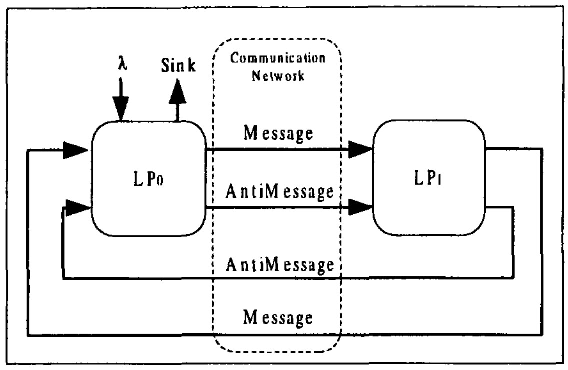

Fig. 6. LPs configuration for the computing system

As discussed early in this paper three types of parameters are required to instrument the model. The parameters related to the synchronisation mechanism are those of the basic Time Warp. As the parameters related to the platform, a cluster of IBM RS6000 workstations and a 16Mbps Token Ring were considered. In this case the time required to execute an event is about 1.208 ms and the time consumed in the transmission of a message (or antimessage) is about 1.4 ms. Two cases were considered for the application parameters, as shown in Table 1. In both cases the probability p is 50%.

Table 1. Tests' configuration

	Interarrival Time (tu)	Service Time 1 (tu)	Service Time 2 (tu)
Test #1	50	10	20
Test #2	50	10	5

Table 2 shows some of the metrics that can be observed by using the proposed methodology. The "rollbacks" metric represents the number of rollbacks caused by the straggler messages or by antimessages. The "messages issued" metric is the number of messages sent by the LP to remote execution. The "antimessage issued" metric represents the number of antimessages sent by the LP.

Table 2. Metrics

Metrics	Test #1		Test #2	
	LP1	LP2	LP1	LP2
Rollbacks	365	391	384	297
Messages issued	414	354	416	354
Antimessages issued	350	265	344	266

These metrics are able to give an initial idea about the impact of the Time Warp mechanisms on the distributed simulation performance. Other metrics can be observed in the Time Warp simulation mechanism such as the relation between the time spent with the events execution and the time spent with rollbacks execution; time spent with antimessages, etc.

3 Final Remarks and Conclusions

This work shows that the use of Petri Nets is adequate to model the behaviour of a distributed simulation synchronisation mechanism because it allows the easy implementation of the activity concurrence and shows the details of each step on a process activity.

Thus, distributed simulation users have a methodology that makes easy and fast the search for a synchronisation mechanism that presents better adaptation and offers the best performance for the problem under study. An user that wants to apply distributed simulation techniques to study some type of problem must only instrument the various models and verify which synchronisation mechanism offers the best performance, without the need to solve the problem using different approaches for synchronisation. The task of setting parameters to the model takes into consideration the hardware platform features that will be used in the future distributed simulation implementation and the features of the application class related with the problem under study.

This paper showed preliminary results where the basic Time Warp model was used. Future work will show the use of the plug-in model and also consider more than one LP executing in the same processing element.

References

1. Das, S. R. Adaptive Protocols for Parallel Discrete Event Simulations." In Proceedings of the 1996 Winter Simulation Conference. (1996) 186-193.
2. Fujimoto, R. M. Parallel and Distributed Simulation Systems. Wiley Series on Parallel and Distributed Computing, John Wiley & Sons, Inc. (2000).
3. Jefferson, D. R. Virtual Time. ACM Transactions on Programming Languages and Systems, v. 7, n. 3 (1985) 404-425.
4. Marsan, M. A.; Bobbio, A.; Donatelli, S. Petri Nets in Performance Analysis: An Introduction. Lecture Notes in Computer Science (1998) n. 1491: 211-256.
5. Misra, J. Distributed Discrete-Event Simulation. ACM Computing Surveys (1986) v. 18, n. 1: 39-65.
6. Moore, K. E. and Hammer, S. D. ALPHA/Sim Simulation Software Tutorial. In Proceedings of the 1999 Winter Simulation Conference (1999) 289-296.
7. Spolon, R.; Santana, M. J.; Santana, R. H. C. Distributed Simulation, Time Warp and its Variations: Taxonomy and Performance Evaluation Issues. In Proceedings of the 13th European Simulation Multiconferece. (1999) 220-227.

Acknowledgement

The authors would like to thank the Brazilian Foundations Capes, CNPq and FAPESP by the support given to the Distributed Systems and Concurrent Programming Group at the ICMC-USP.

NOTAS DO ICMC

SÉRIE COMPUTAÇÃO

052/2000 BRANCO, K.R.L.J.C.; SANTANA, M.J.; SANTANA, R.H.C.; CALÔNEGO JUNIOR, N. – A parallel programming supporting tool.

051/2000 FRANCÊS, C.R.L.; VIJAYKUMAR, N.L.; SANTANA, M.J.; CARVALHO, S.V. de; SANTANA, R.H.C. – Stochastic statecharts for obtaining performance measurements of a file server model.

050/2000 FARIA,G.; ROMERO, R.A.F. – Incorporating fuzzy logic to reinforcement learning.

049/2000 OLIVEIRA, P.R.; ROMERO, R.F.; NONATO, L.G.; MAZUCHELI, J. – Techniques for image compression: a comparative analysis.

048/2000 LINS, L.; LINS, S.; MORABITO, R. – Na n-tet graph approach for non-guillotine packings of n-dimensional boxes into na n-container.

047/2000 OLIVEIRA JR., O. N.; MARCHI, A.R.; MARTINS, M.S.; MARTINS, R.T.; NUNES, M.G.V. – A critical analysis of the performance of english-portuguese-english MT systems.

046/2000 SOUZA, P.S.L.; SANTANA, M.J.; SANTANA, R.H.C. – Escalonamento de processos: uma contribuição para a convergência da área.

045/2000 MARTINS, R.T.; RINO, L.H.M.; NUNES, M.G.V.; MONTILHA, G.; OLIVEIRA JR., O. N. – An interlingua diming at communication on the Web: how language-independent can it be?

044/99 FELTRIM, V.D.; FORTES,R.P.M. – Evaluation of a reverse engineering method through the application in a hypermedia system.

043/99 PANSANATO, L.T.E.; NUNES, M.G.V. – EHDM: Método para projeto de hiperdocumentos para ensino.