Partitioning Simulation Objects in Distributed Environment

Hristo Valchanov, Nadezhda Ruskova, and Trifon Ruskov

Technical University of Varna, 1 Studentska Str,
9010 Varna, Bulgaria
hristo@tu-varna.bg, ruskova@tu-varna.bg, ruskov@tu-varna.bg

Abstract. The parallel discrete event simulation (PDES) is a main approach to improve the execution of a simulation by distributing it between multiple processors. The mapping of the simulation entities over the processors is very important for the performance of the overall process of simulation. This paper presents an approach for mapping the simulation objects over distributed environment based on dynamically intelligent prediction algorithm for building their graph of interactions.

Keywords: PDES, simulation partitioning, simulation objects mapping

1 Introduction

PDES allows for acceleration of the modeling process by distributing it among a number of processors. With PDES, the modeled system is presented as a set of subsystems simulated by a number of simulation objects (SO). The SO communicate with one another by exchanging time-stamped messages for occurring events. The simulation correctness requires that the events be processed in the order of their occurrence in time [1], [2].

The simulation objects may be implemented as separate independent processes. Such implementation, however, is ineffective because of the high system overhead on switching of the processes context by the operating system. Simulation effectiveness improvement can be achieved by aggregating the simulation objects into a cluster. Each cluster will perform as an independent process within a computer node.

This paper presents an approach for mapping of the SO into clusters over distributed simulation environment - network of workstations. The approach is based on dynamically intelligent prediction algorithm for building the graph of SO interactions.

2 Related work

Distribution of SO over computing nodes is very important for the efficiency of the overall process of simulation. Numerous systems for PDES [3] provide such control of distribution, which require the user to map manually SO to the appropriate physical processors. This approach appears to be inefficient for simulation of models, containing many SO with high intensity of interaction. There is a need to automate the process of mapping the appropriate SO to the computing nodes.

The selection of a method for distribution of the components of the simulated system into groups (clusters) has an important role for the efficiency of the overall process of simulation. One method is a representation of the simulated system as a graph, which is to be distributed by means of algorithms for graphs partitioning. By this method, the vertices of the graph represent the individual components of the real system, while the edges of the graph represent the interaction between the components. To these edges are assigned weights, representing the amount of communication between the components. The method is relatively easy to implement, because the problem of graph partitioning is well known in the graphs theory [5].

An important problem for this method is the manner of building of the graph of interactions between SO. A possible solution is based on the *critical path analysis* in the process of simulation [3]. The key concept of this method is that if the graph of the parallel program execution (sequence of events) is known, then the critical path provides the least possible time for execution of simulation. The analyzers, presented in the literature [3], [4], require completion of the whole process of sequential simulation for carrying out the analysis of the critical path (i.e. *post-mortem* analysis). This is applicable to sequential simulation, executed within reasonable period of time. In case of simulation models, involving large number of SO and events such requirement will result in too long execution.

Our approach to building of a graph of interaction of SO is based on preliminary sequential simulation, combined with dynamic intelligent analysis of the interactions between the components of the simulation model. The approach consists of two phases. In the first phase the graph of interactions between SO is created. The second phase includes partitioning the formed graph into clusters.

3 Formation the graph of interactions

In an experimental sequential simulator [6] an analyzing component (AC) is integrated. Its purpose is to analyze the interaction between SO in the preliminary execution of a simulation program. Significant difference between analysis described in the literature and presented method is that it focuses solely on the interaction between SO and not on the sequence of simulation events. As a result of the first phase the AC builds an interaction graph $G(V, E)$ between SO, where $V = \{v_i\}$, $i = 1..N$ is the set of graph vertices. This set corresponds to the set of the simulation objects $O = \{o_i\}$, $i = 1..N$. The set of events $E = \{e_i\}$, $i = 1..M$ is represented as edges of the graph and each edge introduces the interaction between two SO. Two vertices v_k and v_q are connected with edge e_{kq} if the corresponding SO o_k and o_q exchange messages about happened events during the simulation.

The formation of the graph of interactions $G(V, E)$ may be presented as a twosteps process: during the first step, the communication pairs (CP) between SO are formed, and during the second step information about the exchange of messages between them is collected. The primary task is to determine the time T_{CP}, required for the formation of all communication pairs (P_{CP}) during the process of sequential simulation. After formation of P_{CP}, the process of sequential simulation (T_{sim}) continues for a specified period of time - $T_{end} = \lambda * T_{CP}$. During this period data about the amount of communications are collected. The value of the parameter λ indirectly determines the accumulation of information exchanged between communication pairs.

The task for determination of time T_{CP} may be formulated as follows:

(1) define a mathematical model of the process of formation of P_{CP}, type $y = f(x)$, where **y** is the number of communication pairs, **x** is the number of processed events, and **f(x)** specifies the increase of the number of CP as a result of processing of events;

(2) using this model, a prediction is made for the moment in which P_{CP} is reached with specific precision □, i.e. to find such value of **x**, for which $y(x)$ is within the limit □ around the unknown value $y(\infty) = P_{CP}$. For the purpose of these studies we assume the limit □ = 98%.

3.1 Experimental obtaining of y(x)

Experimental studies has been carried out on the basis of the benchmark test PHOLD [2] on the presented in [6] a system for distributed simulation (Fig.1). This test is widely used for assessment of the distributed simulation performance.

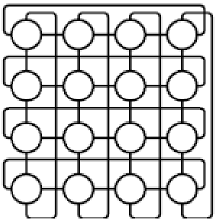
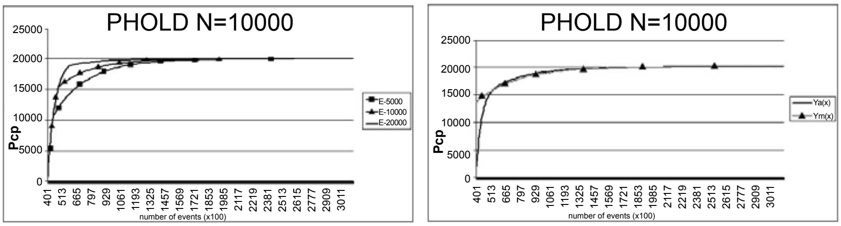


Fig. 1. PHOLD example (N=16).

The test model contains N objects, connected in 2D toroidal network and E events, exchanged between the objects. The dependence has been studied between the number of communication pairs P_{CP} and the number of the processed events during sequential simulation. Fig.2-a shows the obtained characteristics for N=10000.

Based on the experimentally established dependencies, as a model for mathematical description of the variation of P_{CP} may be accepted the differential equation of first degree

$$T * y'(x) + y(x) = K * u(x)\ . \tag{1}$$



(a) (b)

Fig. 2. Experimental *(a)* and approximated *(b)* dependencies.

$$y(x) = K^*(1 - e^{-\frac{x}{T}}) . \quad \textbf{(2)}$$

Fig.2-b shows the results from approximation for different groups of experiments. The conclusion is that the built functional dependences feature relatively good coincidence with the experimental dependences. For the determination of T_{CP} is important only the sector of approximation of P_{CP} to the limit **K**, so we may assume that the selected mathematical model of variation of P_{CP} is adequate to the data.

3.2 Evaluation of the $\mathbf{Y_{0,98}(x)}$

From (2) for the transient characteristic of the selected mathematical model may be written the expression for the value of $y(x)$, reaching 98% from the specified value $y_{0.98}\,(x_{0.98}) = y\,(x_{0.98})$:

$$y\,(x_{0.98}) = 0.98^*K = K^*(1 - e^{-\frac{x_{0.98}}{T}}) . \quad \textbf{(3)}$$

$$x_{0.98} = -T^*\ln(0,02) . \quad \textbf{(4)}$$

From (4) it is evident that for determination of the moment T_{CP} (as a number of processed events $x_{0.98}$), is necessary to determine the time constant **T** of the mathematical model. It is necessary to find such value of **x**, at which $y(x)$ is within a specified limit (0,02) around an unknown value of $y(\infty)$.

By differentiation of (2) we will obtain

$$y'(x) = -\frac{K}{T} e^{-\frac{x}{T}} . \quad \textbf{(5)}$$

Such equation is written for two values of **x**: $\mathbf{x_1}$ and $\mathbf{x_2}$

$$y'(x_1) = -\frac{K}{T} e^{-\frac{x_1}{T}} ,\, y'(x_2) = -\frac{K}{T} e^{-\frac{x_2}{T}} . \quad \textbf{(6)}$$

Let

$$\beta = \frac{y'(x_1)}{y'(x_2)} = \frac{e^{-\frac{x_1}{T}}}{e^{-\frac{x_2}{T}}} = e^{\frac{1}{T}(x_2 - x_1)} . \quad (7)$$

From where

$$T = \frac{(x_2 - x_1)}{\ln(\beta)} . \quad \textbf{(8)}$$

By replacing in formula (4) we will obtain the sought value

$$x_{0.98} = -\frac{\ln(0,02)}{\ln(\beta)}(x_2 - x_1) . \quad \textbf{(9)}$$

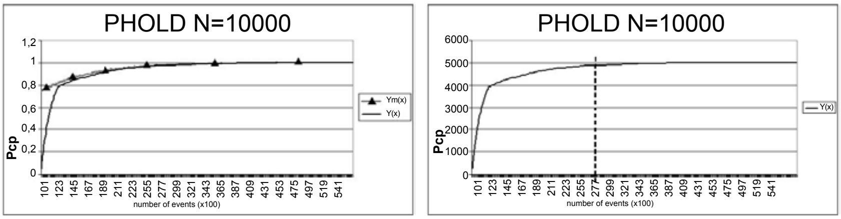
There is a solution at $\ln(\beta) \neq 0$, i.e. $y'(x_1) \neq y'(x_2)$, which is possible to be implemented in practice.

3.3 Experimental evaluation of the proposed approach

The experimental assessment is based on the same test model and the used algorithm includes the following:

1. At every step the current number of communication pairs P_{CP} is followed;
2. After formation of P_{CP} >0, at every step its value is saved and its first derivative is calculated by the method of finite differences;
3. When the 10th report is reached after P_{CP} >0, $\mathbf{x_1}$ and the derivative $y'(x_1)$ is determined;
4. At every next step a check for the achievement of the calculated derivative of the value $0{,}1y'(x_1)$ is made. When it is reached, determine $\mathbf{x_2}$.

For the purpose of the experimental research, it is necessary to finalize the sequential simulation in order to obtain sufficient data about building the model curves. Fig.3-a presents the built functional dependence $\mathrm{Y_m}(\mathrm{x})$ with the calculated parameters of the model (experimental and model curve are normalized against **K**). Fig.3-b shows the moment (as a number of processed events) at which $y(x)$ enters within the desired limit of 98% from the limit value $y(\infty)$: $y(x_{0.98}) = y(285) = 4878$.



(a) (b)

Fig. 3. Model curves (*a*) and the limit reaching (*b*).

In the real case when $y(\infty)$ =5000, $y(x_{0.98})$ = 0,98*5000 = 4900. The difference from the real value as an error is about 1%.

4 Partitioning the graph of interactions

Formally the problem of graph partitioning is defined as follows- a graph $G(V, E)$ with set of vertices V and set of edges E is given. Let V be divided into k subsets $V_1, V_2, \ldots, V_k$ such that:

1. $V_i \cap V_j = \Phi$, $\forall\, i \neq j$, where is an empty set;
2. $\cup_{i=1}^{k} V_i = V$;
3. $|V_i| = \dfrac{|V|}{k}$;
4. The number of edges connecting vertices from different subsets is minimal.

The conditions 1 and 2 determine the division of the graph with number of vertices $|V|$ into k non-overlapping sub-graphs. The remaining two conditions determine the number of vertices in individual sub-graphs to be the same while number of edges between sub-graphs to be minimal.

4.1 Applied method for partitioning

The combinatorial multilevel-based method named *Multilevel k-way* for partitioning of the formed graph G is applied [5]. The choice of the algorithm *Multilevel k-way* is based on the following considerations. First, it incorporates the optimization criterion, very appropriate for simulation in distributed computing environment. The criterion is to minimize the general communication exchange between the computing nodes. Compared to other methods of study, it allows precise graph partitioning at comparatively low computing expenses. Secondly, the algorithm is implemented on the basis of the library METIS [7], which is available as an open source and enables the use of API functions in the applications. The execution of the algorithm *Multilevel k-way* on the formed graph generates a map of SO distribution by computing nodes.

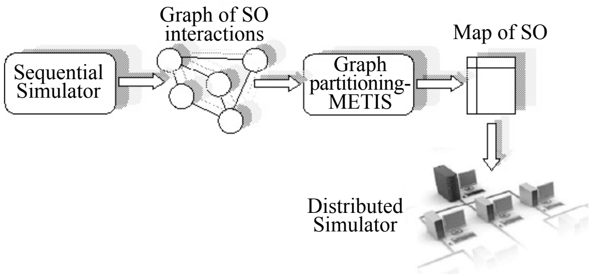


Fig. 4. The SO mapping process.

After completion of the sequential simulation (Fig.4), data are generated for the formed graph of interactions $G(V, E)$ and these data are served as input data for the METIS library. The result is a file, containing the map of distribution of SO between the relevant computing nodes. Once the map is generated, the distributed simulation may start. Based on the information from this map, for each SO subject to creation, the relevant node is assigned.

4.2 Experimental evaluation and results

Experimental assessment is based on the test model PHOLD [2] using SIMOPAL distributed simulation environment [6]. Experiments are carried out in three groups. For the first group, the process of sequential simulation is waited until completion. For the second group of experiments simulation is terminated upon achievement of 90% of the total duration sim T_{sim}. For the third group the proposed method is applied, whereas simulation is terminated upon reaching the time $T_{end} = \lambda * T_{CP}$. Experiments are carried out at different values of λ.

A comparative assessment is made by the following- for each computing node a set Ω of SO, which are assigned to it as a result of the complete process of simulation, and a set $\tilde{\Omega}$, containing SO, assigned to it as a result of the proposed method are formed

$$\bigcup_{j=1}^{n}\Omega_j = \bigcup_{i=1}^{n}\widetilde{\Omega}_i = N\,. \qquad \textbf{(10)}$$

N is the number of all SO in the model, and **n-** the number of computing nodes.

For each set i n i $\widetilde{\Omega}_i$, $i = 1..n$ is determined the maximum ratio $\Psi_i^{\max}$, $i = 1..n$ (in percentages) of the coincidences of its components with each set Ω_i, $j = 1..n$. For each group of distribution by nodes the average maximum ratio is determined

$$\overline{\Psi}_i = \frac{\sum_{j=1}^{n}\Psi_j^{\max}}{i}\ [\%],\ i = 1..n\ . \qquad \textbf{(11)}$$

The aim is on the basis of experimental studies to determine λ at which the value $\overline{\Psi}_i$, $i = 1..n$ will be the highest.

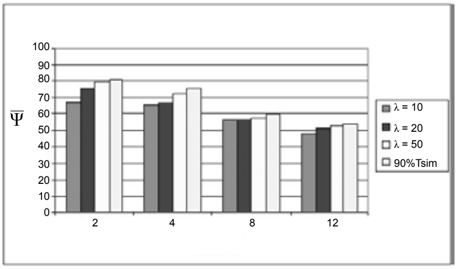


Fig. 5. Experimental results for partitioning.

Fig.5 shows the results obtained from experimental studies. Experiments are carried out by increasing the number of simulated events E. Comparative assessment is made with regard to the results, obtained upon achievement of 90% of the total duration of simulation T_{sim}. As seen from the charts with increasing number of simulated events we observed some reduction in the percentage of matches. This is due to the fact that a larger number of events needed more time to accumulate information about the communication exchange between communication pairs.

Increasing the number of computing nodes N also reduces $\overline{\Psi}$. This is a normal consequence of the increasing of the number of communication channels between communication pairs. It should be noted that this reduction is lower with a larger number of computing nodes. This indicates the correctness of *Multilevel k-way* choice in terms of its resistance to growing the size of the simulation.

As λ increases, it is logical that coincidence of sets $\overline{\Psi}$ increases too, due to the increase of information about the communication exchange between SO. It is very important to note the obtained result of coincidence upon achievement of 90% of the total time for sequential simulation T_{sim}. As it is evident from the graphics, regardless of the number of computing nodes and the number of simulated events, this proportion is within very close limits to the results in the cases of $\lambda > 20$. On this basis we may assume that choosing λ with values higher than 20, will allow obtaining distribution of SO over the computing nodes, which are good enough upon the initial start of distributed simulation.

5 Conclusions and future work

In this paper we proposed an approach to the partitioning of a graph of interaction of SO, which key concept is based on preliminary sequential simulation, combined with dynamic prediction analysis of the interaction between the components of the simulation model. The results show that the proposed approach has efficiency for simulation models, characterized by high dynamics of scheduled events between SO.

The purpose of a future work will be to study the period, necessary to prolong the process of sequential simulation after reaching the required limit of 98% P_{CP}, as well as to study the efficiency of different methods for division of the graph of interactions G.

Acknowledgements. This paper is financed by project "Creative Development Support of Doctoral Students, Post-Doctoral and Young Researchers in the Field of Computer Science", BG 051PO001-3.3.04/13 EUROPEAN SOCIAL FUND 2007-2013, OPERATIONAL PROGRAMME "HUMAN RESOURCES DEVELOPMENT".

References

1. Banks, J., J. S. Carson, II, B. L. Nelson: Discrete-Event System Simulation (5th ed.), Upper Saddle River, Prentice Hall (2009)
2. Fujimoto R.M.: Parallel Discrete Event Simulation. CACM, vol.33, pp. 41-52 (1990)
3. Juhasz Z, Turner S, Gerzson M.: A Trace-based Performance Prediction Tools for Parallel Discrete Event Simulation, In: Applied Informatics, part 3, pp. 338-343 (2002)
4. Carothers C., Fujimoto R.: Efficient Execution of Time Warp Programs on Heterogeneous, NOW Platforms, IEEE Trans. on PADS, v.11 n.3, pp.299-317 (2000)
5. Schloegel K., Karypis G., Kumar V.: Parallel Multilevel Algorithms for Multi-constraint Graph Partitioning. LNCS, vol.1900, pp. 296-310 (2000)
6. Valchanov H., Ruskova N., Ruskov T.: Distributed Simulation over Network of Workstations, In: CompSysTech'2006, pp. IIIB.25-1-IIIB.25-6 (2006)
7. METIS: A family of multilevel partitioning algorithms, http://www.userscs.umn.edu/karypis/90/index.html