Synchronization Aspects of The Optimistic Parallel Discrete Event Simulation Algorithms

Liliia Ziganurova[1,2] and Lev Shchur[1,2]

[1] Scientific Center in Chernogolovka, 142432, Chernogolovka, Moscow region,
[2] National Research University Higher School of Economics, 101000, Moscow
E-mails: ziganurova@gmail.com, levshchur@gmail.com

Abstract. We study synchronization aspects in parallel discrete event simulation (PDES) algorithms. Our analysis is based on the recently introduced model of virtual times evolution in an optimistic synchronization algorithm. This model connects synchronization aspects with the properties of the profile of the local virtual times. The main parameter of the model is a "growth rate" $q = 1/(1 + b)$, where b is a mean rollback length. We measure the average utilization of events and the desynchronization between logical processes as functions of the parameter q. We found that there is a phase transition between an "active phase", i.e. when the utilization of the average processing time is finite, and an "absorbing state" with zero utilization, vanishing at a critical point $q_c \approx 0.136$. The average desynchronization degree (i.e. the variance of local virtual times) grows with the parameter q. We also investigate the influence of the sparse distant communications between logical processes and found that they do not change drastically the synchronization properties in the optimistic synchronization algorithm, which is the sharp contrast with the conservative algorithm [1]. Finally, we compare our results with the existing case-study simulations.

Keywords: discrete event simulation, parallel discrete event simulation, PDES, optimistic algorithm, small-world

1 Introduction

Parallel discrete event simulation (PDES) [2] is a powerful tool of programming on high-performance computing systems [3]. It is widely used for modeling complex systems in computer science, engineering, physics, economics, and society [4]. The main advantage of PDES is that it is highly scalable by construction, for example, PDES simulator ROSS [5] is able to scale up to 1.9 million cores running a synthetic PHOLD model [6]. Even though the ideas of the method emerged around 40 years ago [7], the study of the PDES is still important nowadays. State-of-the-art and research challenges in the area of parallel simulation can be found in recently published papers [8, 9]. The study of PDES is going in many directions: studying different properties of PDES models [10], optimization of simulation kernels [11–14], different usage of PDES, e.g. internet of things [15], etc. In this paper we investigate properties of optimistic PDES algorithm, using a model of evolution of local virtual times.

The idea of PDES is that the physical system is simulated as a set of subsystems, which communicate with each other by time-stamped messages. The subsystems are mapped on programming objects, or logical processes (LPs). The logical process executes a sequential subprogram with its own local state variables and its own local virtual time (LVT) on some processing elements (nodes, processors, cores, or threads). During the simulation, the LPs interact by sending time-stamped event messages to each other. Each LP has an input and output queues of events. The received event messages, which are waiting for execution, are stored in the input queue, and the messages, which must be sent to other LPs, are located in the output queue. The messages in both queues are sorted by timestamp order. The simulation process goes as follows: each LP takes the first message from its input queue, executes an event, changes its local state and local virtual time, and sends messages to other LPs, if necessary. LPs works in parallel independently, without global synchronization. The simulation result will be correct (i.e. as if the simulation was sequential) if all the events have been executed by all LPs in correct non-decreasing timestamp order. In PDES the synchronization is carried out by each LP by the analysis of the values of timestamps in the queue, according to some synchronization protocol. There are three classes of the synchronization protocols: conservative, optimistic, and Freeze-and-Shift (FaS) protocol [2, 16, 17]. PDES algorithm can be classified using the mapping of the algorithm onto the partial differential equation describing the surface growth [18] and analyzing the boundary conditions [17]. In this scheme, the open boundary conditions correspond to the optimistic algorithm, the periodic boundary conditions correspond to the conservative algorithm, and the fixed boundary conditions correspond to the FaS algorithm.

In conservative synchronization, only secure events are allowed to be processed. The event is called secure, if we are sure that during the execution of this event the LP will not receive a message with a lower timestamp. This is usually implemented by using block-resume mechanisms, such that flags, semaphores, etc. The optimistic algorithm, in contrary, allows causality violations but provides a rollback mechanism for causality recovery.

All of the synchronization algorithms have their pros and cons and should be used according to the available computational facilities and the particular knowledge on the simulated system. For example, conservative synchronization is a better choice for systems with good lookahead information, i.e. the information on the minimal time between two dependent events. The conservative algorithm is easier to implement, but it generally works more slowly than the optimistic one. Realization of the optimistic algorithms are more complex, but usually, have better performance and can be used for a wider class of models [19].

Our research is focused on the study the synchronization properties of the optimistic PDES algorithm on different communication networks via the analysis of the local virtual time profile (Fig. 1). Such an approach was introduced for conservative synchronization algorithm in [20], and extended to other PDES algorithms and topologies in works [1, 21–27]. The approach provides rather a theoretical point of view on the synchronization PDES algorithms and allows to make general predictions about their behaviors. Moreover, the model can be attributed to the models of surface growth in physics, which allows using a rich instrument of statistical physics for the analysis of our model.

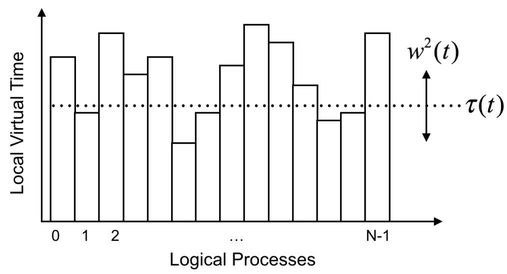

Fig. 1. A snapshot of local virtual time profile at a simulation step t. The LPs have their values of the LVT. $\tau(t)$ is a local virtual time averaged over all logical processes and $w^2(t)$ is an average squared width of the profile.

The paper is organized as follows. In Section 2 we describe the model of evolution of LVTs in optimistic PDES algorithm. Section 3 provides the simulation results. In Section 4 we discuss the results and compare them with the existing case-studies of PDES models.

2 Model description

In this section, we describe a model [27] of evolution of LVT profile in optimistic PDES. We do not simulate any particular optimistic synchronization algorithm. Instead, we focused on the behavior of the local virtual times simulating the *model* of the optimistic PDES algorithm. There is a one-to-one correspondence between the LVT profile and the synchronization aspects of the optimistic algorithm. The average speed of the profile reflects the utilization of events or the effectiveness of processors load, and the profile width can be thought as a measure of desynchronization degree between LPs. The desynchronization shows the deviation of LVTs from the average between all LPs. A small deviation from the average time indicate that the LPs work at more or less equal pace, and none of the LPs are too ahead or behind from the others, while a high value of the desynchronization degree implies that some LPs are ahead and some of the LPs are behind, what increases a probability of causality violations and makes the leading processes to wait for the actual information from the lagging processes.

As a consequence, the average efficiency of the simulation slows down because of high desynchronization between LPs.

In optimistic PDES algorithms, the LPs are allowed to execute events independently without synchronization. At this stage, the LVT profile is growing freely. When the causality of computations is violated, i.e. some LP receives a message with a timestamp lower, than its LVT, the mechanism of rollback is run. This LP changes its LVT and state variables to the value when the receiving the erroneous message would be safe. After that, all sent messages must be "unsent". This is done by sending so-called anti-messages – the same messages but with the opposite sign. When a message and its anti-message occur in the same queue, they annihilate. It is clear, that one rollback can cause an avalanche of rollbacks. When the processing of rollbacks has been finished, the LVT profile will be in average lower and flatter, since some of the LPs changed their LVT to the lower values.

We simulate this process as follows. First, we set a communication topology. The communication topology determines the dependencies of LPs and can be presented as a graph, where vertices represent the LPs and edges represents the dependencies between the LPs (Fig. 2). The dependent LPs exchange by the messages and the independent ones do not communicate. Then we initiate an array of LVTs (i.e. the LVT profile) and update it according with the rules described below in this chapter. During the simulation, we calculate the observables: the average speed and the average squared width of the LVT profile. We use the following assumptions:

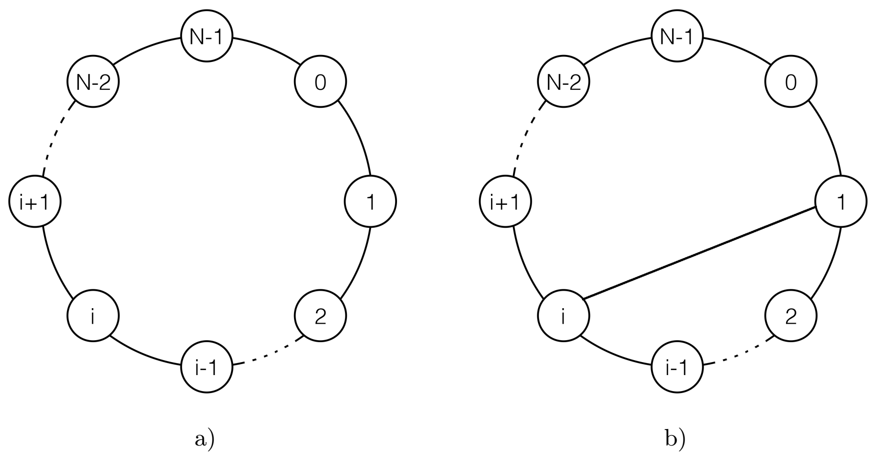

Fig. 2. a) Regular communication topology. In this topology all communications are local. b) Small-world communication topology. Besides the local communications, there are additional distant communications. The LP_1 and LP_i are distant processes, connected by a communication link.

1. The communication topology is known and fixed in advance: it is known, which LPs exchange by messages and which LPs are independent.
2. Times between two events are random variables exponentially distributed with the mean 1.
3. Sending time and receiving time are equal (i.e. there is no communication overhead).
4. The causality violation may occur with equal probability at any LP.
5. If LP_i depends on the information from several LPs, the causality violation on the LP_i may be caused by any of those LPs with equal probability.
6. The number of rollbacks is a discrete random variable exponentially distributed with the mean b.

1. Setting the communication topology. We consider a system of N logical processes connected into a communication graph. We are interested in two communication topologies: regular and small-world because they reflect interconnections in real systems [28]. In regular topology, the LPs are arranged into a ring such that each LP depends on the two neighboring LPs: one at the left and one at the right (Fig. 2a). In small-world topology, we add a small amount of communications between distant LPs above the ring (Fig. 2b). Small-world topology is characterized by the low value of the average shortest path – it scales logarithmically with the system size, while in regular networks the average shortest path growth linearly with N. It was shown in [24] that these distant communications significantly enhance the synchronization between LPs in conservative synchronization algorithm.

The amount of distant communications is controlled by the parameter p. The total amount of distant communications is equal to pN. The case of $p = 0$ corresponds to a regular network.

2. Simulation of evolution of the LVT profile. When the communicational graph is determined, we start to simulate an evolution of LVTs. We begin with a flat LVT profile $\tau_i(t = 0) = 0$, $i = 1, 2, ., N$, where N is a number of LPs, ant t is one simulation step in our model. We assume that one simulation step consists of two stage: 1) simulation of profile growth, and 2) simulation of rollbacks. At the first stage each LP_i increases its LVT by random value η_i, drawn from the Poisson distribution: $\tau_i(t + 1) = \tau_i(t) + \eta_i, i = 1, \dots, N$.

To simulate a rollback, we randomly choose a LP_i and compare its LVT τ_i with the value of LVT τ_r of one of its neighbors LP_r, chosen with equal probability. If $\tau_i > \tau_r$, we set τ_i equal to τ_r. We repeat this action several times, assuming that the number of rollbacks is a random value drawn from the Poisson distribution with the mean b. The actions described above constitute one simulation step t. The full simulation consists of M simulation steps.

3. Calculation of the observables. After one simulation step t (increasing LVT profile + rollbacks) we calculate the observables:

1. The average height of the LVT profile $\tau(t)$ – an arithmetical mean of all LVTs at simulation step t.

$$\tau(t) = \frac{1}{N}\sum_{i=1}^{N}\tau_i(t).$$

2. The average speed of the profile $u(t)$ – an increment of the average height of the profile after one simulation step.

$$u(t) = \tau(t+1) - \tau(t)$$

3. The average squared width of the profile $w^2(t)$ – a statistical variance of LVTs from the mean value $\tau(t)$.

$$w^2(t) = \frac{1}{N}\sum_{i=1}^{N}[\tau_i(t) - \tau(t)]^2.$$

The described algorithm of evolution of LVT profile in optimistic PDES in pseudocode looks as follows:

```
Set parameters N, M, p, b
Create a communication graph
for t := 0; t < M; t++ do
    for i := 0; i < N; i++ do τ_i(t) += η_i
    k = Poisson(b)
    for j := 1; j < kN; j++ do
        Choose random LP_m
        Choose random neighbour of LP_m LP_r
        if τ_m(t) > τ_r(t) then τ_m(t) = τ_r(t)
    Calculate observables
```

3 Simulation results

We investigate the average speed and the average squared width of the LVT profile, which reflect such properties of the optimistic algorithm as the utilization of events and desynchronization between LPs, accordingly. We performed our simulation on regular and small-world topologies, varying the parameter p from 0 to 0.1. Number of LPs is fixed to $N = 10^4$, number of the simulation step M changes from 10^3 to 10^5. We also introduce a parameter $q = 1/(1 + b)$, where b is a mean rollback length. The parameter q controls a growth rate of the profile and changes in our models from 0 to 1. We conduct the simulation using random number generation library RNGAVXLIB [29] and average the results over 1000 independent realizations of the models with fixed parameters.

The average speed on a regular topology. The average speed of the profile shows, how fast the LPs utilize the events. In our model the LVT profile growth with constant velocity, therefore we omit time dependence in the next formulas. We found, that the average speed u decreases with the parameter q, and when q approaches to some critical value q_c, the speed becomes equal to 0. Such behavior can be explained by a high amount of rollbacks, which do not let the profile of LVT grow (q is reversely proportional to the number of rollbacks).

We approximate the average speed u as a function of q by the following formula:

$$u(q) = u_0 \cdot (q - q_c)^\nu \tag{1}$$

The results of the fit of the data to the expression (1) are: $u_0 = 1.26(2)$, $q_c = 0.136(1)$, $\nu = 1.78(2)$. The behavior of the speed shows phase transition between an "active phase" (when $u > 0$), and "pinned phase" (when $u = 0$). Such behavior reminds a transition in directed percolation models [30]. It is interesting, that the critical exponent ν is also close to the critical exponent of directed percolation universality class.

The average speed on a small-world topology. When the LPs are connected into a small-world communication network, the behavior of the average speed slightly changes. The critical point q_c shifts to the right, when the parameter p increases. It happens, because the number of dependencies between LPs is increasing with p, therefore the probability of longer rollback avalanche is higher. The critical exponent ν also grows with p.

The average squared width on a regular topology. The average squared width of the LVT profile characterizes the degree of desynchronization between LPs. The width grows in a power-law manner with time t and then saturates. The saturation time and saturation value is higher for the larger parameter q. It is explained by the fact, that the number of rollbacks, in this case, is low, therefore the LVT profile grows freely.

The average squared width on a small-world topology. The behavior of LVT profile in the optimistic PDES algorithm does not exhibit qualitatively changes, when the underlying topology changes from regular to a small world, as it happens in the conservative algorithm [1]. The average squared width also grows with time and the parameter q but decreases slightly with the concentration of long-range connections p. As in the conservative algorithm, the additional communication links make the LVT profile smoother, i.e. the LPs work more synchronized. However, the difference between regular and small-world topologies in the optimistic algorithm is not so significant, because the mechanism of rollback reduces the difference between LVTs, even in the absence of additional communications between LPs.

4 Discussion

We analyzed the synchronization properties of optimistic PDES algorithm on regular and small-world communication topologies, using the model [27]. The model was introduced for the optimistic algorithm with only local interactions between logical processes. The results of our study have shown that the model is also applicable to the qualitative predictions of the synchronization properties of the optimistic algorithm with more general types of communication topology.

We found, that there is a critical point, at which the growth of the LVT profile stops, i.e. the utilization of events becomes zero. It means, that for systems with a high probability of rollbacks the optimistic algorithm would not be efficient. We also compared the results on regular and small-world topologies and found that the additional distant communications do not play such an important role as in the conservative PDES algorithm, where the synchronization was significantly better on small-world topology than on the regular topology [1].

For the application of our model to the real simulations, it is necessary to find an analogy between the parameters of the simulated systems and the parameter q of the present model. It is also possible to compare our results with the existing case-studies of various PDES models.

Paper [31] summarizes the profile data captured from 22 discrete-event simulation models from 4 simulators: NS-3 [32, 33], ROSS [5], WARPED2 [34], and Simian [35]. The research focuses on the communication properties of events exchanges between the LPs, namely, LP connectivity, betweenness centrality, and modularity. The analysis of LP connectivity has shown that in most models the LPs have either a fixed amount of connections (regular topology) or 1-8 connections in some proportion (as in small-world topology). The tendency of LPs to communicate with only a few other LPs makes the models good for parallel execution. The same LP connectivity is seen in our model as well, however, we cannot provide a detailed description of betweenness centrality and modularity of communication graphs in our model.

In [36] the performance of PDES is studied on ROSS simulator running PHOLD model on Knights Landing Processor. It was shown that the number of submitted events is decreasing with the fraction of remote events (event-messages passing between different cores). However, the simulation performance scales linearly with the number of cores, if each LP is assigned to its core, and the fraction of remote events is less than 10%. In our simulations we studied the topologies with a small fraction of remote connections (from 0.1 to 10%), and also found that they slightly slow down the performance (i.e. the average speed of the LVT profile) in both, the conservative and the optimistic algorithms. At the same time in the conservative algorithm they drastically enhance the synchronization (i.e. the average squared width of the LVT profile).

Another analogy between the observations in [36] and our results can be drawn regarding the interval of Global Virtual Time (GVT) update. The GVT is a minimum value among all LVTs. The state variables of LPs are stored only until the GVT. Smaller GVT interval requires less state information to be kept, but increase the overhead of GVT calculations. On the other hand, the rollback length is shorter, therefore the calculation of rollbacks goes faster. The interval of GVT computation has some similarity to the parameter q of our model.

The average speed of the profile in our model has values from 0 to 1. It can be compared with the average utilization of events in [10] varying from 0.47 for an epidemic model to 0.0043 for traffic model, and down to $5 \cdot 10^{-5}$ for wireless network model on running ROSS [5] and WARPED2 [34] simulators.

In the future, we plan to perform case-study simulations of the existing PDES models and establish relationships between the parameters of the real parallel discrete-event simulations and the parameters of our models.

5 Acknowledgements

The work has been done within the research theme 0236-2019-0001.

References

1. Ziganurova, L., Shchur, L. N.: Synchronization of conservative parallel discrete event simulations on a small-world network. Physical Review E, **98**, 022218 (2018). doi:10.1103/PhysRevE.98.022218
2. Fujimoto, R.M.: Parallel discrete event simulation. Communications of the ACM **33**, 30-53 (1990). doi: 10.1145/84537.84545
3. Bailey, D.H., David, H., Dongarra, J., Gao, G., Hoisie, A., Hollingsworth, J., Jefferson, D., Kamath, C., Malony, A., Quinian, D.: Performance Technologies for Peta-Scale Systems: A White Paper Prepared by the Performance Evaluation Research Center and Collaborators. White paper, Lawrence Berkeley National Laboratories (2003). doi: 10.2172/15004540
4. Tropper C.: Parallel Discrete-Event Simulation Applications. Journal of Parallel and Distributed Computing **62**, 3, 327-335 (2002). doi: 10.1006/jpdc.2001.1794.
5. Carothers, C.D., Bauer, D., Pearce, S.: ROSS: A high-performance, low-memory, modular Time Warp system. Journal of Parallel and Distributed Computing **62**, 1648-1669 (2002). doi: 10.1016/S0743-7315(02)00004-7
6. Barnes Jr, P. D., Carothers, C. D., Jefferson, D. R., and LaPre, J. M.: Warp speed: executing time warp on 1,966,080 cores. In Proceedings of the 1st ACM SIGSIM Conference on Principles of Advanced Discrete Simulation, 327-336 (2013). doi: 10.1145/2486092.2486134
7. Jefferson D., Fujimoto R.: A Brief History of Time Warp. In Advances in Modeling and Simulation, 97-134 (2017). Springer, Cham. doi: 10.1007/978-3-319-64182-9_7
8. Balci O., Fujimoto R. M., Goldsman D., Nance, R. E., and Zeigler, B. P.: The state of innovation in modeling and simulation: the last 50 years. In 2017 Winter Simulation Conference (WSC), 821-836 (2017). IEEE. doi: 10.1109/WSC.2017.8247835
9. Fujimoto R., Bock C., Chen W., Page E., and Panchal, J. H.: Research challenges in modeling and simulation for engineering complex systems. Springer. (2017). doi: 10.1007/978-3-319-58544-4
10. Wilsey P.A.: Some Properties of Events Executed in Discrete-Event Simulation Models. In: Proceedings of the 2016 annual ACM Conference on SIGSIM Principles of Advanced Discrete Simulation, 165-176 (2016). ACM, New York. doi: 10.1145/2901378.2901400

11. Ross C. J., Carothers C. D., Mubarak M., Ross R. B., Li J. K., and Ma K. L.: Leveraging Shared Memory In The Ross Time Warp Simulations. In 2018 Winter Simulation Conference (WSC) 3837-3848. IEEE. (2018). doi: 10.1109/WSC.2018.8632333
12. Eker A., Williams B., Mishra N., Thakur D., Chiu K., Ponomarev D., and Abu-Ghazaleh N.: Performance Implications of Global Virtual Time Algorithms on a Knights Landing Processor. In 2018 IEEE/ACM 22nd International Symposium on Distributed Simulation and Real Time Applications (DS-RT), 1-10. IEEE. (2018). doi: 10.1109/DISTRA.2018.8600923
13. Masko L., and Tudruj M.: Application global state monitoring in optimization of parallel event-driven simulation. Concurrency and Computation: Practice and Experience, **e5015** (2018). doi:10.1002/cpe.5015
14. Knopov P., and Pardalos P.M., eds. Simulation and optimization methods in risk and reliability theory. Nova Science Pub Incorporated, 2009.
15. D'Angelo G., Ferretti S., and Ghini V.: Simulation of the Internet of Things. In 2016 International Conference on High Performance Computing and Simulation (HPCS) 1-8. IEEE. (2016) doi: 10.1109/HPCSim.2016.7568309
16. Jefferson, D.R.: Virtual time. ACM Transactions on Programming Languages and Systems (TOPLAS) **7**, 404-425 (1985). doi: 10.1145/3916.3988
17. Shchur, L.N., Novotny, M.A.: Evolution of time horizons in parallel and grid simulations. Physical Review E **70**, 026703 (2004). doi: 10.1103/PhysRevE.70.026703
18. Kardar, M., Parisi, G., Zhang, Y.C.: Dynamic scaling of growing interfaces. Physical Review Letters, **56**, 889 (1986). doi: 10.1103/PhysRevLett.56.889
19. Jefferson D. R., and Barnes Jr P. D.: Virtual time III: Unification of conservative and optimistic synchronization in parallel discrete event simulation. In Proceedings of the 2017 Winter Simulation Conference, 55. IEEE Press. (2017). doi: 10.1109/WSC.2017.8247832
20. Korniss G., Toroczkai Z., Novotny M.A. and Rikvold, P.A.: From massively parallel algorithms and fluctuating time horizons to nonequilibrium surface growth. Physical Review Letters **84**, 1351 (2000). doi: 10.1103/PhysRevLett.84.1351
21. Shchur L.N., Shchur, L.V.: Relation of Parallel Discrete Event Simulation algorithms with physical models. Journal of Physics: Conference Series **640**, 012065 (2015). doi: 10.1088/1742-6596/640/1/012065
22. Shchur L., Shchur L.: Parallel Discrete Event Simulation as a Paradigm for Large Scale Modeling Experiments. In: Selected Papers of the XVII International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL 2015), Obninsk, Russia, October 13-16, 2015, pp. 107-113 (2015). http://ceur-ws.org/Vol-1536/
23. Alon U., Evans M.R., Hinrichsen H., Mukamel D.: Roughening transition in a one-dimensional growth process. Physical Review Letters, **76**, 2746 (1996). doi: 10.1103/PhysRevLett.76.2746
24. Guclu H., Korniss G., Novotny M. A., Toroczkai, Z., Racz, Z.: Synchronization landscapes in small-world-connected computer networks. Physical Review E **73**, 066115 (2006). doi: 10.1103/PhysRevE.73.066115
25. Ziganurova L., Shchur L.: Properties of the Conservative Parallel Discrete Event Simulation Algorithm. Lecture Notes in Computer Science **10421**, 246-253 (2017). doi: 10.1007/978-3-319-62932-2_23
26. Shchur L., Ziganurova L.: Simulation of Virtual Time Profile in Conservative Parallel Discrete Event Simulation Algorithm for Small-World Network. Lobachevskii Journal of Mathematics, **38**, 5, 967-970 (2017). doi:10.1134/S1995080217050316

27. Ziganurova L., Novotny M.A., Shchur L.N.: Model for the evolution of the time profile in optimistic parallel discrete event simulations. In: Journal of Physics: Conference Series **681**, 012047 (2016). doi: 10.1088/1742-6596/681/1/012047
28. Watts D.J., Strogatz S. H.: Collective dynamics of small-world networks. Nature **393** (6684), 440, (1998). doi: 10.1038/30918
29. Guskova M.S., Barash L.Y., Shchur L.N.: RNGAVXLIB: Program library for random number generation, AVX realization. Computer Physics Communications **200**, 402-405 (2016). doi: 10.1016/j.cpc.2015.11.001
30. Odor G.: Universality classes in nonequilibrium lattice systems. Reviews of modern physics, **76** (3), 663, (2004). doi:10.1103/RevModPhys.76.663
31. Crawford P., Eidenbenz S. J., Barnes P. D. and Wilsey P. A.: Some properties of communication behaviors in discrete-event simulation models. 2017 Winter Simulation Conference (WSC), Las Vegas, NV, 1025-1036 (2017). doi: 10.1109/WSC.2017.8247852
32. Henderson T. R., Lacage M., Riley G. F., Dowell C., and Kopena J.: Network Simulations with the ns-3 Simulator. SIGCOMM demonstration **14**, 527. (2008)
33. Riley G. F., Henderson T. R.: The ns-3 Network Simulator, 1534. Springer. (2010).
34. Weber D.: Time warp simulation on multi-core processors and clusters. Master's thesis, University of Cincinnati, Cincinnati, OH (2016).
35. Santhi N., Eidenbenz S., and Liu J.: The Simian Concept: Parallel Discrete Event Simulation with Interpreted Languages and Just-In-Time Compilation. In Proceedings of the 2015 Winter Simulation Conference, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 30133024. Piscataway, New Jersey, USA: Institute of Electrical and Electronics Engineers, Inc. (2015) doi:10.1109/WSC.2015.7408405
36. Williams B., Ponomarev D., Abu-Ghazaleh N., and Wilsey P.:Performance characterization of parallel discrete event simulation on knights landing processor. In Proceedings of the 2017 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation 121-132. ACM. (2017). doi: 10.1145/3064911.3064929