# Properties of The Conservative Parallel Discrete Event Simulation Algorithm

Liliia Ziganurova[1,2] and Lev Shchur[1,2,3]

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

**Abstract.** We address question of synchronisation in parallel discrete event simulation (PDES) algorithms. We study synchronisation in conservative PDES model adding long-range connections between processing elements. We investigate how fraction of the random long-range connections in the synchronisation scheme influences the simulation time profile of PDES. We found that small fraction of random distant connections enhance synchronisation, namely, the width of the local virtual times remains constant with increasing number of processing elements. At the same time the conservative algorithm of PDES on small-world networks remains free from deadlocks. We compare our results with the case-study simulations.

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

## 1 Introduction

Modern high performance systems consist with hundreds of thousands of nodes, which in turn may have many CPUs, cores, and numerical accelerators. The development of hardware architecture influences on the development environments (programming models, frameworks, compilers, libraries, etc.). They now need to deal with a high level of parallelism and can solve difficulties arising from system heterogeneity [1].

In the paper we discuss synchronisation in one of the methods of large-scale simulation known as parallel discrete event simulation (PDES) [2]. The method is widely used in physics and computer science, as well as in economics, engineering, and society. The first ideas come about 40 years ago in order to overcome limitation of memory/time resources, and revising of the method is still important nowadays. PDES has a property of good scalability with the physical system size (number of objects) as well as with the hardware size (number of nodes, cores, and level of the hyper-threadings).

PDES allows to run one single discrete event simulation task on the number of processing elements (PE), which physically can be nodes, or CPUs, or cores, or threads depending on the particular system architecture. The system being simulated is divided into subsystems which are mapped onto programming objects, or logical processes (LPs). Logical process is a sequential subprogram executed by some PE.

The system changes its state at some discrete moments of time, which are usually Poisson arrivals. The changes are called *discrete events*. The events generates messages with timestamps which are saved in the output queue and sent to other LPs. Received messages are stored by LPs in their input queues. LPs during the simulation maintain a loop, sequentially taking the event with the lowest timestamp from the input queue, executing it, and communicating with other LPs if necessary. The communication between LPs goes exclusively via time-stamped messages. It is important that LPs do not use any shared memory. Synchronisation process is done locally by the analysis of the values of timestamps of messages in the queue. When LP processes an event, it updates its own local virtual time (LVT) to the time of the processed event. Each LP evolves independently in time and there is no global synchronisation in the simulation process. Since the dynamic is asynchronous some synchronisation protocol is required. There are three groups of such protocols: conservative, optimistic and FaS [2–4].

In the paper we investigate the performance and scalability properties of the conservative PDES algorithm. In conservative algorithm it is assumed that all dependencies between LPs must be checked before every portion of computations in order to preserve causality of the computations. The performance of conservative algorithm depends on the communication network: the more dependencies in the system the lower speed of the computation. We study the influence of long-range communication links on the synchronisation and performance of PDES conservative algorithm. We build a simplified model of the evolution of LVT profile. The model allows to measure local time variance and average speed of the utilisation of processing times by LPs. The observables are then mapped onto synchronisation aspects of PDES scheme.

The paper is organised as follows. In the next section we describe a background of the problem. Section 3 provides detailed information about the model under consideration. The results of our simulation are given in Section 4. The discussion and further work are presented in Section 5.

## 2 Models of Evolution of LVT Profile in PDES

In this section we describe the general approach to investigation of synchronisation in PDES algorithms and review main results in this area.

*Conservative PDES model on regular networks.* Model of evolution of times in PDES conservative algorithm is proposed in [5]. Authors consider communication scheme with only nearest-neighbour interactions, which is equivalent to one-dimensional system with periodic boundary conditions. It was found in simulation that evolution of time profile reminds the evolution of a growing surface which is known from literature in physics does belong to Kardar-Parisi-Zhang-like kinetic roughening [6]. This analogy provides a cross-disciplinary application of well-known concepts from non-equilibrium statistical physics to our problem. More details on the relation of PDES algorithms with physical models can be found in [7, 8].

Synchronisation of the parallel schemes can be described using this analogy. *Efficiency* of parallel implementation can be defined as a fraction of the non-idle processing elements. This fraction exactly coincides with the density of local minima in the growing model. It is shown in [5] that in the worst-case scenario the efficiency of the PDES algorithm remains nonzero as the number of PEs goes to infinity.

*Freeze-and-Shift PDES model.* The conservative PDES is proved to be free from deadlock, and efficiency is about 1/4, on average, i.e., at least one PE out of four is working at any given time. In [4] an alternative synchronisation algorithm of PDES was proposed. It is based on i) the extension of the PDES concept on the hierarchical hardware architecture, including multi-core and hyper-threadings and ii) using analogy of the evolution of time-profile interface with the physical models of surface interface growth. In the late case classification of the boundary conditions leads to the classification of possible PDES algorithms. Authors give a way to increase utilisation by giving each node a large portion of LPs which are processed by threads running on the same CPU. The LPs within one CPU communicate conservatively, whereas LPs from different CPUs communicate according to either conservative, or optimistic scheme, or scheme with fixed LVT on the boundary LPs (those which can communicate with LP in the neighbouring CPUs). In the last case CPUs do not communicate with other CPUs for some time interval window (the frozen part of the algorithm), and after that time the message exchange is implemented as part of the memory shifting between CPUs (the shift part of the algorithm). The algorithm is therefore called Freeze-and-Shift (FaS).

*Optimistic PDES models on regular networks.* Model of evolution of time profile in optimistic PDES algorithm is introduced in [9]. Dynamical behaviour of the optimistic PDES model is quite different from the conservative PDES model. The optimistic model corresponds to another surface growing model and demonstrates features of the roughening transition and directed percolation [10].

*Conservative PDES models on small-world like networks.* All models described above consider PDES algorithms with short-range connections. The idea of studying the model with other type of communication topology is proposed in [11]. Authors investigate the behaviour of the local virtual time profile on a small-world like network [12]. The network is build as a regular one-dimensional lattice with additional long-range connections randomly wired above it. The links are used dynamically, i.e. at each time step additional synchronisation check between distant neighbours is made with some probability $p$. Small value of the $p$ significantly improves synchronisation. Variance between local times becomes finite, while utilisation decreases just slightly.

In present paper we revise the approach of [11], considering more realistic topology, and compare with the result of [11]. In addition we compare our results with the case-study [13].

## 3 Model Definition

We build a model of *evolution of LVTs profile* for analysis of the desynchronisation processes in conservative PDES algorithm. The PEs are said to be synchronised when the differences between LVTs stay finite with the simulation process. The efficiency of the algorithm is measured as a number of PEs working at a given moment of time. It reflects the load of the processing elements (CPU, core, or thread). When the efficiency of the algorithm is strictly greater than zero, one can say that the algorithm is deadlock free. These properties of the synchronisation algorithm can be extracted from the analysis of the LVT profile.

The model is constructed on the *small-world communication networks* [12]. Long-range connections in addition to short-range reflect real systems properties. For example, computer networks, social networks, electric power grid, and network of brain neurones are known to be small-world networks. Additional communication links between distant nodes also enhance synchronisation of simulation.

First we build a communication topology. For simplicity of the model we assume that each PE does process only one LP. Nodes in the communication graph represent PEs, and edges represent dependencies between them. Each PE has its local variable $\tau$, which is the value of LVT. The set of all LVTs is stored as an array. Two observables are computed: the efficiency $\langle u \rangle$ and the profile width $\langle w^2 \rangle$. We run the simulation program with different set of parameters $N$ and $p$. Finally we take an average over multiple samples.

*Topology.* Denote number of processing elements as $N$. We build a small-world topology of PEs using the parameter $p$ – the fraction of random long-range connections. First we connect all PEs into a regular one-dimensional lattice (equivalent to the ring) and then add $pN$ random long-range connections. Each edge is chosen only once. The result is a communication graph of $N$ nodes and $N(1+p)$ edges stored as adjacency list.

*Initialisation.* We set the parameter $p$ of the network to some value from 0.002 to 0.01, build a topology, and set all local times to zero: $\tau_i(0) = 0$, $i = 1..N$.

*Simulation.* We are interested in evolution of LVTs profile in *conservative* algorithm. We assume that only those PEs, whose current time is lower than the time of their neighbours (i.e. the PEs which it is connected with), may proceed with computations. These PEs are called *active.* Such scheme guarantees that causality will be preserved [2].

In our model we implement an ideal scheme of message passing. At each time step $t$ every LP broadcasts the message with the time stamp equal to its LVT to all LPs connected with it. We assume that time needed for message distribution is negligibly small, so there is no difference between sending and receiving time. The PEs who have received only messages with higher time stamp than their LVT, may proceed. These PEs have minimal LVT among their neighbours.

At each simulation step $t$ we find PEs with the lowest LVTs among their neighbouring nodes and increment local time of those PEs by an exponentially distributed random amount:

$$\tau_i(t+1) = \begin{cases} \tau_i(t) + \eta & \text{if } \tau_i(t) \le \tau_K(t), \\ \tau_i(t) & \text{otherwise,} \end{cases} \tag{1}$$

where $\eta$ is a random value drawn from the Poisson distribution, $K$ is a set of all PEs which are connected to $i$-th PE by local or long-range communication links, and $i = 1..N$.

After updating array of LVTs the observables are computed, and PDES goes to the next simulation cycle.

*Observables.* We compute two essential features of the model: the efficiency $\langle u \rangle$, which is equivalent to the average utilisation of the algorithm, and the width of the profile $\langle w^2 \rangle$, i.e. the variance of local virtual times, which is associated to the desynchronisation of PEs.

1. The *efficiency* (utilisation) of the algorithm is equivalent to the density of local minima of the LVT profile. The efficiency shows how many PEs is working at a given moment of time. In basic conservative scheme on a ring topology (when each PE is connected with exactly two neighbours) the efficiency is approximately 1/4. The figure is derived analytically from the observation of all possible combination of LVTs of neighbouring PEs. Numerical result is equal to 0.24641(7) [5]. The number shows that only approximately one quarter of all PEs are working at a given moment of time and other three quarters are idling.
   We calculate the efficiency of the algorithm $\langle u \rangle$ as an average fraction of active PEs at each time step:

$$\langle u \rangle = \frac{\langle \overline{N_{activePE}} \rangle}{N}. \tag{2}$$

   The underlined average is taken over all time steps and $\langle \cdot \rangle$ states for the average over independent 1500 runs with fixed parameters $N$ and $p$.
2. The *width* (variance) of the LVT profile shows the average spread between local virtual times. If the width remains constant during the simulation, then PEs are well synchronised. The increasing of the profile width corresponds to the growing of the desynchronisation with simulation time.
   The width of the LVT profile is calculated according the formula below:

$$\langle w^2(N,t) \rangle = \left\langle \frac{1}{N} \sum_{i=1}^{N} [\tau_i(t) - \overline{\tau}(t)]^2 \right\rangle, \tag{3}$$

   where $\overline{\tau}(t) = \frac{1}{N} \sum_{i=1}^{N} \tau_i(t)$ is the mean value of the time profile.

In our simulation program we use random number generation library RNGAVXLIB [14]. We run program on the Manticore cluster using MVAPICH2 [15].

## 4 Results

We are interested in scalability properties of the synchronisation of conservative PDES model with the long-range connections. We perform simulation of conservative PDES model on the ring topology with long-range connections. We simulate systems of size $N$ (number of PEs) varying from $N = 10^3$ to $N = 10^5$, and for number of values of fraction $p$ for the long-range connections. Note that $p = 0$ corresponds to the basic conservative model with only short-range connections studied in [5].

The main results are: 1) efficiency of the algorithm remains finite and slightly reduces with adding long-range connections $p$; 2) profile width for any $p$ grows with the system size; 3) profile width saturates for system sizes larger than $10^4$; 4) degree of desynchronisation depends logarithmically with $p$.

*The efficiency.* We observe that for any system size the average density of local minima $\langle u(t)\rangle$ monotonically decreases as a function of time and approaches a constant. The constant depends on the fraction of random connections $p$ and system size $N$. For small $p$ the utilisation of events reduces slightly. The small-world-synchronised simulation scheme maintains an average rate greater than zero. For example, for $p = 0.01$ it is $\langle u\rangle = 0.22137(7)$, while for basic conservative scheme $\langle u_0\rangle = 0.24641(7)$ [5].

The efficiency $\langle u\rangle$ has nonlinear dependence on the parameter $p$. It is possible to fit utilisation dependence on $p$ by expression:

$$\langle u(p)\rangle = u_0 - A(N)p^{B(N)}. \tag{4}$$

The coefficient $A$ and the exponent $B$ depend on the system size, and can be fit using logarithmic or exponential dependencies. Using logarithmic fit we obtain $A = 0.078(3) + \frac{0.345(9)}{\log N}$ and $B = 0.092(3)) + \frac{1.26(1)}{\log N}$. Using power-law fit we obtain $A = 0.08(2) + \frac{0.253(5)}{N^{0.12(3)}}$ and $B = 0.24(1) + \frac{1.14(5)}{N^{0.21(1)}}$. We could not choose which fit is better.

Finally, we found that as $N$ goes to infinity, $\langle u(p)\rangle = u_0 - 0.078(3)p^{0.092(3)}$ if we approximate $A$ and $B$ with logarithm, or $\langle u(p)\rangle = u_0 - 0.08(2)p^{0.24(1)}$ in the case of the power-law approximation for the coefficients $A$ and the exponent $B$.

*The width.* We observed that the profile width grows as $\langle w^2(t)\rangle \sim t^{2\beta}$ and saturates at some time $t_\star$ reaching the value $\langle w^2_\infty\rangle$. We measured the growth exponent $\beta$ for each combination of the parameters $N$ and $p$. For large systems ($N > 10^4$) exponent $\beta$ becomes almost constant. The asymptotic value of $\beta$ is found to behave logarithmically with $p$

$$\beta = -0.137(4) - 0.162(1)\ln(p). \tag{5}$$

We remind that without long-range connections ($p = 0$) it is $\beta = 1/3$.

It is interesting, that during the simulation on more then ten thousand PEs the desynchronisation of PEs will grow equally fast for systems of any sizes.

For large systems synchronisation depends only on the amount of long-range connections. The profile width approaches a constant value $\langle w^2_\infty\rangle$ with growing system size. In contrary, in the basic short-range conservative scheme ($p = 0$) the width is increasing with the system size as $\langle w_\infty^2 \rangle \sim N^{2\alpha}$, $\alpha = 0.49(1)$. The topology with additional distant connections allows the simulation of large systems to preserve the same degree of synchronisation.

Since the small-world conservative PDES scheme progresses with positive rate and the profile width becomes *finite* in the limit of infinitely many PEs, one can say that the conservative algorithm with long-range connections is fully scalable.

## 5 Discussion and future work

In the paper we present analysis of the synchronisation in conservative PDES algorithm on the small-world networks.

Paper [13] presents detailed results of the case-study simulations of different models. Two optimistic simulators were used: ROSS [16] and WARPED2 [17]. Simulation results of three models were reported: traffic model, wireless network model, and epidemic model. Average utilisation $\langle u \rangle$ varied from 0.47 for epidemic model to 0.0043 for traffic model, and down to $5{\cdot}10^{-5}$ for wireless network model.

We guess that the results of case-study [13] can be explained in part by the concept of small-world network with varying parameter $p$. To answer this question in details it is necessary to perform case-studies of the mentioned models measuring quantities, which can be mapped on the parameters of our model.

In addition, it is interesting to investigate properties of the *optimistic* algorithm of PDES on small-world networks provided with the comparison with the results of case-studies.

## 6 Acknowledgements

This work is supported by grant 14-21-00158 of the Russian Science Foundation.

## References

1. 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)
2. Fujimoto, R.M.: Parallel discrete event simulation. Communications of the ACM **33**, 30-53 (1990). doi: 10.1145/84537.84545
3. Jefferson, D.R.: Virtual time. ACM Transactions on Programming Languages and Systems (TOPLAS) **7**, 404-425 (1985). doi: 10.1145/3916.3988
4. 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
5. 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 (200). doi: 10.1103/PhysRevLett.84.1351

6. Kardar, M., Parisi, G., Zhang, Y.C.: Dynamic scaling of growing interfaces. Physical Review Letters, **56**, 889 (1986). doi: 10.1103/PhysRevLett.56.889
7. 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
8. 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/
9. 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
10. 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
11. 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
12. Watts, D.J., Strogatz, S.H.: Collective dynamics of “small-world” networks. Nature **393**, 440-442 (1998). doi: 0.1038/30918
13. 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, pp. 165-176 (2016). ACM, New York (2016). doi: 10.1145/2901378.2901400
14. 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
15. MVAPICH: MPI over InfiniBand, Omni-Path, Ethernet/iWARP, and RoCE. http://mvapich.cse.ohio-state.edu
16. 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
17. Weber, D.: Time warp simulation on multi-core processors and clusters. Master’s thesis, University of Cincinnati, Cincinnati, OH (2016).