

PROPERTIES OF THE PARALLEL DISCRETE EVENT SIMULATION ALGORITHMS ON SMALL-WORLD COMMUNICATION NETWORKS

Liliia Ziganurova^{1,2,a} Lev Shchur^{2,3,b}

¹ Science 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-mail: ^a ziganurova@gmail.com, ^b lev.shchur@gmail.com

Synchronization aspects in the method of large-scale simulation, known as parallel discrete event simulation (PDES), analyzed using the models of the time profile evolutions. Time profile is formed with the local virtual times of the processes. Time profile evolution in the simplest cases reminds the growth of the surface in the models of statistical physics. This simplest case considers only local dependences, the message exchange with the nearest neighbors. In the real simulations, the exchange can be with any of the processes, and we can consider the communication network of messages forms the small-world network. We found the enhancement of synchronization with the growing average shortest path of the small-world network. At the same time, the utilization drops down, although on the moderate level. We present the preliminary results of our study. The work is supported by grant 14-21-00158 of the Russian Science Foundation.

Keywords: parallel discrete event simulation, conservative algorithm, optimistic algorithm, virtual times, local times profile, synchronization, small-world networks

© 2018 Liliia Ziganurova, Lev Shchur

1. Introduction

The computational platforms for simulations have undergone dramatic changes in recent years. A number of cores in modern supercomputers has grown up to millions, and computers have become rather heterogeneous as they consist of a number of CPUs, GPGPUs, and numerical accelerators [1]. Simulation now faces new computational challenges, concerning the complexity of computing platforms [2]. One of the methods which may be efficiently used in large-scale simulation in parallel discrete event simulation (PDES) [3].

Parallel discrete event simulation method can be viewed as an interacting set of sequential simulations of parts of the simulated system, without changing the original dynamics of the simulated system [4]. Each sequential simulator is called *logical process* (LP) and may be executed by a single thread, core, CPU, or node. Changes in the system state occur at discrete, typically irregular, points in simulation time. These changes are therefore called *discrete events*. The LPs interact by exchanging timestamped messages, which represent events, scheduled by one LP to another. The timestamp represents a point in simulation time at which the state change occurs.

The synchronization protocols [5] are based on the concept of *virtual time* [6]. Each LP has its *local virtual time* (LVT), which changes when LP handles an event or sends a message. The collection of LVT of all LP constitutes *LVT profile*. The LVT profile is growing during the execution of the simulation program. Understanding the dynamics of LVT profile helps in studying of properties of synchronization algorithms. An average speed of the LVT profile characterizes the average utilization of the processing elements. The average width of the profile (standard deviation) can be considered as desynchronization between LPs. Moreover, simulation of LVT profile evolution for systems of different sizes allows studying the scalability properties of the algorithms [7].

Model of the conservative algorithm was studied in references [7, 8]. The goal of the paper is to study the properties of the conservative and optimistic PDES algorithms on small-world (SW) networks [9]. SW networks are characterized by the small length of the average shortest path and by the large value of the clustering coefficient. In [10] we found that the behavior of the LVT profile mostly depends on the average shortest path in the network, and clustering coefficient does not play a big role. For the simplest SW model, the long-range links are added above the 1-dimensional ring (Figure 1).

2. The models

Consider a system of N logical processes. The communication between LPs is arranged in small-world topology, schematically shown in Figure 1. We create the SW topology as follows. We build a ring of N LPs (i.e. each LP has exactly two neighbors), and then randomly add pN links between different LPs above the regular lattice. The parameter p is responsible for the fraction of long-range connections. The so build communication graph can be described by adjacency matrix D , such that $D(i, j) = 1$, if LP i and LP j depend on each other's state, and $D(i, j) = 0$, if LP i and LP j are independent.

Evolution of LVT profile $\{\tau_i\}, i = 1..N$ is simulated in this communication topology. We start from a flat profile: $\{\tau_i(0) = 0\}, i = 1..N$, and update it each computational step. After each update we calculate observables. We average our results over 1000 independent realizations. In our simulation program, we use library RNGAVXLIB [11] for the random number generation.

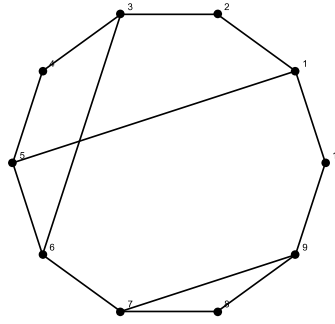


Figure 1. An example of the small-world communication network with the number of LPs $N = 10$ and fraction of long-range links $p = 0.3$

2.1. The model for conservative algorithm

Conservative synchronization algorithm is based on the idea of preserving all “insecure” actions of LPs, i.e. the actions, which potentially may violate the causality of computation. It is usually implemented with using “block-resume” constructions. The simplest conservative scheme is to let LP processing events, only if its local virtual time is less or equal to the LVT of its neighbors. If the condition is satisfied, this LP is called active, otherwise, the LP is idle. Assuming, that events are Poisson arrivals, one may describe the evolution of LVT profile by the following rule.

$$\tau_i(t+1) = \begin{cases} \tau_i(t) + \eta_i, & \text{if } \tau_i \leq \{\tau_j(t)\}_{D(i,j)=1}, \\ \tau_i(t), & \text{otherwise,} \end{cases} \quad (1)$$

where η_i is a random variable drawn from the Poisson distribution. At each update step t we calculate the following observables:

1) Average speed:

$$u(t) = \left\langle \frac{1}{N} \sum_{i=1}^N [\tau_i(t) - \tau_i(t-1)] \right\rangle \quad (2)$$

2) Average width:

$$w^2(t) = \frac{1}{N} \langle \sum_{i=1}^N (\tau_i(t) - \bar{\tau}(t))^2 \rangle, \quad (3)$$

where $\bar{\tau}$ is the average value of LVT: $\bar{\tau}(t) = \sum_{i=1}^N \tau_i(t)$. The triangular brackets here and further in the paper mean the average over independent copies of the simulation programs.

2.2. The model for optimistic algorithm

The fundamental synchronization primitive in the optimistic algorithm is a rollback, instead of blocking constructions [12]. In the optimistic algorithm, any LP may send messages to other LPs at any time, without preservation of the causality. If the LP receives a message “from the past”, the algorithm detects an error and recovers the events which were processed prematurely.

The recovery is usually implemented by passing anti-messages. Message and anti-message carry the same information and differ only by the sign. If both, message and anti-message, meet in the same queue, they annihilate. For example, if an error has occurred at LP_i, but it has already sent erroneous messages to LP₂, after detecting an error, the LP_i sends anti-message to LP₂, and rolls back its LVT. But LP₂, in its turn, may have already processed the erroneous message and sent the events to others LPs. This causes the avalanche of rollbacks.

Simulation of the behavior of the optimistic algorithm consists of two parts: simulation of forwarding evolution of the LVT profile, and simulation of rollbacks. During the forward evolution, each LP_{*i*} updates its LVT τ_i , increasing it by value η_i , taken from the Poisson distribution with unit mean:

$$\tau_i(t + 1) = \tau_i + \eta_i, \text{ for } \forall i = 1..N. \quad (4)$$

To simulate rollbacks, we make each PEs decrease its LVT to the value of LVT of one of its neighbor, chosen with equal probability. We assume that the avalanche length (i.e. number of rollbacks) follows the Poisson distribution with mean b . The algorithm of simulation rollbacks is presented by the following pseudocode:

```

k = Poisson(b)
For j = 1 to kN do:
    Choose random LPi
    Choose random neighbor of LPi LPr
    If  $\tau_i(t) > \tau_r(t)$ , then  $\tau_i(t) := \tau_r(t)$ ,
    
```

At each update step t , consisting of forwarding evolution and rollbacks, we calculate the average speed (2) and the average time profile (3). The parameter of the model is $q = 1/(1 + b)$, the ratio of the forward sub-steps to the total number of sub-steps, constituting the one step in t .

3. Simulation results

We show how adding a small fraction of long-range links affects properties of the algorithm.

3.1. Results of simulation for conservative algorithm

The speed u of the profile for a conservative algorithm in case of regular topology ($p = 0$) is approximately equal to $1/4$ [5]. It means that only one-fourth of the LPs are working at one moment of time, and other $3/4$ are waiting. Such low efficiency is a cost for safety of computations: LPs work only when they are insured that the causality will not be violated. This is achieved by checking the dependencies between LP before every portion of computation. Therefore, when we add more communication links ($p > 0$), the speed of the profile reduces (Figure 2, left). We found that for the infinitely large system ($N \rightarrow \infty$), $u = u_0 - p^{0.306(4)}$.

The width of the profile grows with time and saturates at value $\langle w_\infty^2 \rangle$. In case of regular topology ($p = 0$) this saturation value $\langle w_\infty^2 \rangle$ increases with the number of LPs as $N^{2\alpha}$, with $\alpha = 1/2$. Adding even very small amount of long-range communication links between LPs suppress the desynchronization: $\langle w_\infty^2 \rangle$ does not increase any more with the number of LPs (Figure 2, right).

So, the conservative algorithm on SW networks becomes *fully scalable*.

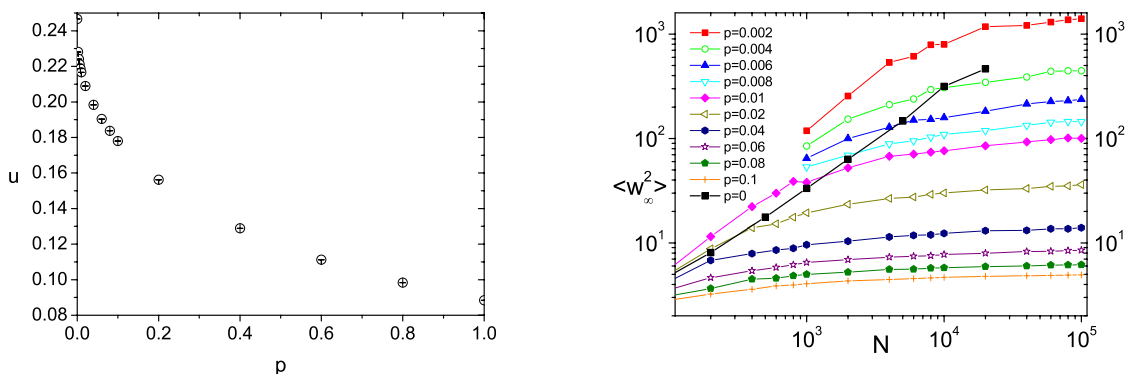


Figure 2. (Left) average speed u as a function of parameter p for the system of 10^6 LPs; (right) average steady-state width $\langle w_\infty^2 \rangle$ as a function of the number of LPs for different value of parameter p

3.2. Results of simulation for optimistic algorithm

We found the average speed of the profile increases as power of q :

$$u(q) = u_0(q - q_c)^\nu \quad (6)$$

The critical value q_c and the exponent ν does depend on the parameter of SW network p . For the regular lattice ($p = 0$) $q_c = 0.143(1)$ and $\nu = 1.66(1)$, and they increase with the number of long-range connections. There is no qualitative difference in the behavior of the width profile (Figure 3, left).

The behavior of the average width also does not change qualitatively with changing the communicational topology. The desynchronization increases with time and saturates at value $\langle w_\infty^2 \rangle$. The saturation value $\langle w_\infty^2 \rangle$ growth with the parameter q (Figure 3, right), that means that the higher the progress rate, the less the LPs are synchronized.

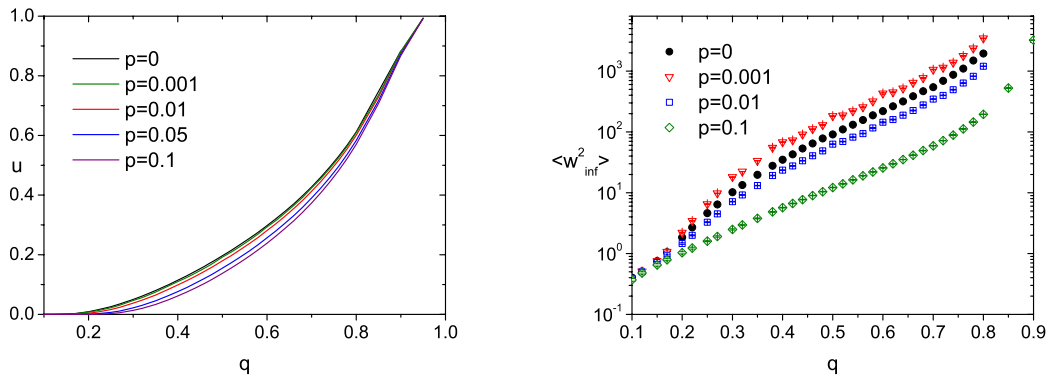


Figure 3. Average speed u (left) and average steady-state width $\langle w_\infty^2 \rangle$ (right) as a function of parameter q for the system of 10^4 LPs and for values of parameter p

4. Conclusion

The main results of the simulation of LVT profile for conservative algorithms on SW networks are: 1) the speed of the profile is positive and reduces slowly with p ; 2) the profile width is finite in the limit of infinite number of LP, i.e. the LPs are well synchronized.

The main results of the simulation of LVT profile for optimistic algorithms on SW networks are: 1) the behavior of the speed and the width of the LVT profile qualitatively does not change with introducing of the long-range communicational links; 2) the average speed of the profile increases as the power of the parameter q ; 3) there is a critical value q_c , below which the progress of simulation fluctuates around zero; 4) the desynchronization degree grows with the progress rate q .

References

- [1] Carothers C., et al. Computational Challenges in Modeling and Simulation. // Research Challenges in Modeling and Simulation for Engineering Complex Systems. Springer, Cham, 2017: P. 45-74.
- [2] Fujimoto R. M. Research challenges in parallel and distributed simulation. // ACM Transactions on Modeling and Computer Simulation, 2016: Vol. 26. – No. 4. – P. 22.
- [3] Fujimoto R.M. Parallel discrete event simulation. // Comm. of the ACM, 1990: Vol. 33. P. 30-53.
- [4] Lubachevsky B. D. Efficient parallel simulations of asynchronous cellular arrays. // Complex Systems 1, 1987: pp. 1099-1123
- [5] Shchur L. N., and Novotny M. A. Evolution of time horizons in parallel and grid simulations. // Physical Review E, 2004: Vol. 70. – No. 2. – p. 026703.

- [6] Jefferson D. R. Virtual time. // *ACM Transactions on Programming Languages and Systems*, 1985: Vol. 7. – No. 3. – pp. 404-425.
- [7] Korniss G., et al. From massively parallel algorithms and fluctuating time horizons to nonequilibrium surface growth. // *Physical Review Letters*, 2000: Vol. 84. – No. 6. – p. 1351.
- [8] Korniss G., et al. Suppressing roughness of virtual times in parallel discrete-event simulations. // *Science* 2003: Vol. 299. – No. 5607 – pp. 677-679.
- [9] Watts D. J., and Strogatz S. H. Collective dynamics of “small-world” networks. // *Nature*, 1998: Vol. 393. – No. 6684. – p. 440.
- [10] Ziganurova L., and Shchur L. N. Synchronization of conservative parallel discrete event simulations on a small-world network. // *Physical Review E*, 2018: Vol. 98. – No. 2. – p. 022218.
- [11] Guskova, M.S., Barash, L.Y., Shchur, L.N. RNGAVXLIB: Program library for random number generation, AVX realization. // *Computer Physics Communications*, 2016: Vol. 200. – pp. 402-405 (2016). doi: 10.1016/j.cpc.2015.11.001.
- [12] Jefferson D, and Fujimoto R. M. A Brief History of Time Warp. // *Advances in Modeling and Simulation*. Springer, Cham, 2017: pp. 97-134.