Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 119 (2017) 139–146

Procedia Computer Science

www.elsevier.com/locate/procedia

6th International Young Scientists Conference in HPC and Simulation, YSC 2017, 1-3 November 2017, Kotka, Finland

Adaptive load balancing of distributed multi-agent simulations on heterogeneous computational infrastructures

Oksana Severiukhina*, Pavel A. Smirnov, Klavdiya Bochenina, Denis Nasonov and Nikolay Butakov

ITMO University, Saint Petersburg, Russia

Abstract

Simulation of the agent-based model has several problems related to scalability, the accuracy of reproduction of motion. The increase in the number of agents leads to additional computations and hence the program run time also increases. This problem can be solved using distributed simulation and distributed computational environments such as clusters and supercomputers. The model objects must be divided into different processes and calculations to be able to be executed in parallel. This paper presents the research on an algorithm to balancing of computational load. The algorithm is based on a well-known genetic algorithm and performs optimization of matching between the model structure formed by multiple interconnected executable blocks of a distributed programming implementation of the model and network structure of the computational environment. Efficient placement of the model graph's nodes to heterogeneous computational ones leads to improvement in overall performance of the simulation and reducing of execution time.

Peer-review under responsibility of the scientific committee of the 6th International Young Scientist conference in HPC and Simulation

Keywords: agent-based modeling; distributed computing; load-balancing

1. Introduction

Modeling of large-scale and complex systems is an important challenge. Such simulations can find applications in different fields. Reproducing natural crowd movement is one of them. One of the wide accepted methods of modeling and studying various social processes and phenomena, such as the movement of people, transport, etc., are agent

* Corresponding author. *E-mail address:* oseveryukhina@gmail.com

1877-0509
Peer-review under responsibility of the scientific committee of the 6th International Young Scientist conference in HPC and Simulation
10.1016/j.procs.2017.11.170

models. One of the traits which is common for such simulations is a requirement for the high amount of computational resources. These resources are often spent on parameters investigation (for example, grid search) and repetition in order to accumulate enough statistical data. Each instance of such simulation consists of multiple blocks that can be run on different computational nodes and make use of distributed programming frameworks (for example, written using MPI).

Meanwhile, the structure of modern computational environments such as clusters and supercomputers may assume heterogeneity in network connections. Often their nodes are connected with one of the well-known typical topologies such as fat tree [1], butterfly [2], dragonfly [3] and others. The most widely accepted of them is a fat tree. This topology has a complex hierarchical structure with several levels that differ by bandwidth and point-to-point throughput for nodes using them in different segments (e.g., leafs of the fat tree). The topology of the computational network may also affect the overall computations performance, especially in case of federated infrastructures. For example, simulations may be scale by several remote clusters connected via the internet. However, additional network-specific information is required for better clusters distribution and agents' exchange. Information about bandwidths from different clusters of racks may be used for better agents' distribution and a decrease of non-optimal communication overheads. Authors [4] use genetic algorithm for workflow scheduling in case of heterogeneous distributed environment.

Individual instances of such models for efficient execution may be placed on nodes from the same network segments to avoid overheads implied by transferring between different segments. But often there are too many instances and they cannot be placed only in one segment and must use resources from few ones. Also, if there are other users who use this computational environment such situation can happen due to their workload.

From the other side, the structure of the application implementing such a model depends on the configuration of this model. Nowadays, multi-agent modeling is used for the different scientific area: pedestrian movement transport modeling, information distribution, modeling biological processes and so on. Each of the areas has its own characteristics, which must be taken into account during modeling this process. One of the important parameters is the environment in which the agents are located. A particular case of such an environment is a rectangular area. This can occur when modeling traffic on the highway, blood cells in the body [5] or movement of the crowd [6].

This can be illustrated with the following example. In Figure 2, one can see a model for simulation of agent movements in the city of Saint-Petersburg. In the model, individual districts and municipalities are connected between themselves and form a graph where edges are weighted. Their weights express the expectation of volume of agents that will be transferred between districts. A district itself can be viewed as a cluster of agents. However, it can be divided further into subclusters to obtain, even more, fine-grained structure, in this work we limit splitting of the model to district level only.

The important thing is that both graphs – topology of the network in a computational environment and cluster-based structure of the model being computed – introduce the notion of nodes proximity and require carefully adjusted matching of one graph to another to be efficiently executed. Neighborhood nodes in the graph of the model with high volumes of transferred agents should be placed on the topology graph on computational nodes with high bandwidth while weakly connected can be moved to different regions in the network topology. These conditions lead to an optimization problem of two graphs matching.

This paper contributes with:

1. An algorithm of matching model structure to network topology based on genetic algorithm.
2. Experimental study of the proposed algorithm's efficiency.

2. Related works

Wang and Lees distinguish several approaches to organizing a distributed computing scheme for the agent model of pedestrian movement [7]: centralization of modeling space; duplication of modeling space; division of space into zones, including agents located in this zone; independent separation of space and agents.

In the first case, one process is responsible for the state of the modeling space; agents are distributed among the remaining processes. This approach has a significant drawback: when modeling large-scale territories with a large number of agents, the bottleneck problem arises, associated with a large number of calls to the process responsible for the state of the modeling space. The second approach solves this problem by creating additional copies of the modeling