# Distributing Agent-Based Simulation

**Richárd O. Legéndi and Attila Szabó**

The exponential growth of computational capacity of personal computers played an important role in the spread of computational modeling. Simulation techniques such as Agent-Based Modeling (ABM) are becoming a common practice, where enormous runs claim vast computational resources because of the need of precise and robust large scale experimental results. Multi-core systems (like clusters, grids, clouds, or even desktop PCs) offer multiplied resources at a relatively low cost, but effective utilization of these systems is still a challenge for software technology.

In the context of ABM distribution, the dynamics of the communication network between agents play an important role. When the parts of a distributed model are highly interconnected, the overhead of network communication can easily reduce the performance of the multi-core system. Gulyás et al. defined a classification of six types (based on common communication schemes), and introduced a solution to distribute models having static communication networks [1]. Scheutz and Schermerhorn defined adaptive algorithms for parallel execution of "spatial AMBs", where agents move in a spatial environment [2]. Gilbert et al. used a peer-to-peer infrastructure to simulate the emergence of artificial societies [3]. Yamamoto et al. created a simulation environment that enables to host millions of agents on a system of workstations connected with a high performance network [4].

In this paper we study the performance of some existing tools of distributed software execution by running agent-based simulations on a cluster of computers. We measure the effectiveness and practical scalability of these frameworks by implementing selected models available at the the OpenABM Consortium's on-line model repository [5] using ProActive [6], JavaSpaces [7], and other Java RMI-based tools [8]. We also seek for effective implementation techniques and practices to support large-scale simulations, those exceeds the limitations of a single computer (e.g. by using several gigabytes of memory). Our goal is to reach a competitive simulation speed compared to sequential execution by keeping the cluster's communication overhead at a relatively low level.

## Acknowledgements

This work was supported by a grant from the Hungarian Government (TÁMOP-4.2.1/B-09/1/KMR-2010-0003).

**References**

[1] Gulyás, L. et. al.: Templates for Distributed Agent-Based Simulations on a Quasi-Opportunistic Grid, *EMSS* (2008).

[2] Scheutz, M., and Schermerhorn, P.: Adaptive Algorithms for the Dynamic Distribution and Parallel Execution of Agent-Based Models, *Journal of Parallel and Distributed Computing*, 66, 1037-1051., (2005).

[3] Gilbert, N. et al.: Emerging Artificial Societies Through Learning, *Journal of Artificial Societies and Social Simulation 9* (2006).

[4] Yamamoto, G., Tai, H., and Mizuta, H.: A platform for massive agent-based simulation and its evaluation, *In Proceedings of the 6th international Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '07)* (2007).

[5] Model Archive, *OpenABM Consortium's site. Web* (7. May, 2010.)

[6] Baude, F. et al.: GRID COMPUTING: Software Environments and Tools, *Jose C. Cunha and Omer F. Rana (Eds), Springer Verlag* (2006).

[7] Freeman, E., Arnold, K., and Hupfer, S.: Javaspaces Principles, Patterns, and Practice, *1st. edition Addison-Wesley Longman Ltd.* (1999).

[8] Wollrath, A., Riggs, R., and Waldo, J.: A Distributed Object Model for the Java System (2010).