A Platform for Massive Agent-based Simulation and its Evaluation

Gaku Yamamoto
IBM Research,
Tokyo Research Lab.
1623-14, Shimotsuruma,
Yamato-shi, Kanagawa, Japan
+81-46-215-4639
yamamoto@jp.ibm.com

Hideki Tai
IBM Research,
Tokyo Research Lab.
1623-14, Shimotsuruma,
Yamato-shi, Kanagawa, Japan
+81-46-215-5260
hidekit@jp.ibm.coml

Hideyuki Mizuta
IBM Research,
Tokyo Research Lab.
1623-14, Shimotsuruma,
Yamato-shi, Kanagawa, Japan
+81-46-215-4965
E28193@jp.ibm.com

ABSTRACT
There are many studies on ABS and several frameworks for ABS have already been published. However, there are few frameworks that can enable agent-based simulation using large numbers of agents. We have been developing a Java-based platform for Massive Agent-Based Simulation (MABS) called "Zillions of Agents-based Simulation Environment" or ZASE. The purpose of ZASE is to develop MABS applications on multiple computers. ZASE is designed to host over millions of agents. We introduce ZASE in this paper. We evaluated ZASE for the agent-based auction simulation where the number of agents varied from ten to a million. The results indicate that the number of agents affects the final bid prices and their distributions. Performance measurement results on both an SMP computer environment and a multiple-computer environment are shown.

Keywords
Agent-based Simulation, Massively Multi-Agent System

1. INTRODUCTION
There are many studies on ABS and several frameworks for ABS have already been published [1, 2, 3, 4, 5, 6]. However, a simulation platform for managing large numbers of agents (we call such a simulation a Massive Agent-Based Simulation (MABS) in this paper) has not matured. We have been developing a Java-based platform for MABS. We called the platform "Zillions of Agents-based Simulation Environment" or ZASE in this paper. The purpose of ZASE is to develop MABS applications on multiple computers. ZASE is designed to host over millions of agents.

We introduce ZASE in this paper. We evaluated ZASE for the agent-based auction simulation described in [8]. In the evaluation, we varied the number of agents from ten to a million. This paper presents the simulation results. The results indicate that the number of agents affects the final bid prices and their distribution. Performance measurement results on both an SMP computer environment and in a multiple-computer environment are also shown. An overview of ZASE is introduced in Section 2. An agent-based simulation as an application example and its simulation results are described in Section 3. Performance measurement results are shown in Section 4. Related Work and Conclusions are covered in Sections 5 and 6.

AAMAS'07, May 14–18, 2007, Honolulu, Hawai'i, USA.

2. Overview of ZASE
ZASE is a scalable platform for multi-agent simulations potentially using billions of agents. On the ZASE platform, a simulation environment is divided into multiple runtimes, and they are executed either on a single computer or on multiple computers. Each agent runtime provides a management mechanism for hundreds of thousands or millions of agents. ZASE interconnects multiple distributed processes by means of message communication and integrates them into a single system. ZASE provides simulation developers not only with some basic features such as simulation cycle management and logging but also various essential capabilities required for simulations which use millions of agents. Figure 1 is an outline of the architecture of ZASE. Simulators are executed on simulation runtimes, and agents are executed on agent runtimes. Figure 2 shows two examples of distributed configuration of ZASE.

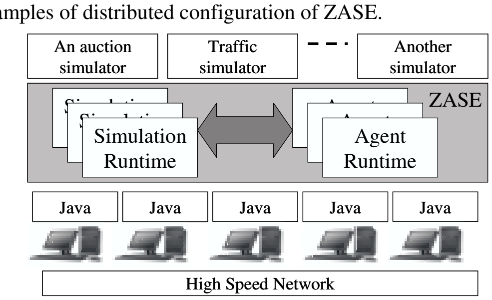
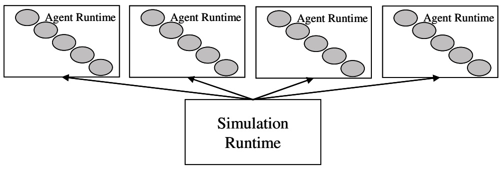


Figure 1. Architectural overview of ZASE

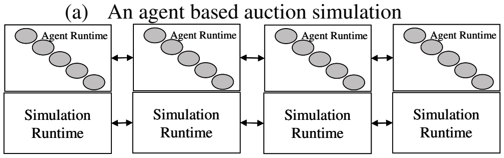
ZASE is based on the agent model of [9]. Each agent of ZASE has internal states, handles messages and updates its own states as needed. Messages are sent by a simulator or by other agents. When a message object is delivered to an agent, a callback method of the agent is called. A returned message object will be sent back to the sender. The ZASE framework provides functions

978-81-904262-7-5 (RPS) ©2007 IFAAMAS

to create and to delete agents. Developers can add services to an agent runtime. Agents can look up the reference to a service and can directly invoke methods of services. A service object also has functions similar to those of agents. Messages are exchanged among agents, simulation runtimes, and agent runtimes. ZASE provides point-to-point messaging and multicast messaging. ZASE also provides a message distribution and aggregation function. When a simulator sends a multicast message to an agent runtime, the agent runtime will distribute the message to agents. An agent runtime aggregates reply messages from agents into an aggregated message and sends the message to a simulator.



(a) An agent based auction simulation



(b) Swarm-type simulation

Figure 2. Examples of simulation environment divisions

3. An Application Example

We applied ZASE to the agent-based auction simulation described in [8]. In the simulation, there are two types of bidder agents: An EarlyBidder or a Sniper. An EarlyBidder starts bidding from when the auction opens and raises the bid price little by little. A Sniper bids at a high price once just before the auction closes. This is a Vickrey auction, so the auction model is that the agent bidding at the highest price wins, but only pays a price equal to the second highest bid. In [8], the number of agents was fixed at ten. We ran six trials for 10^x agents, with x ranging from 1 to 6 (10 to 1,000,000 agents). The ratio of Snipers was set to 30%. We iterated the simulation for each number of agents one thousand times. The simulation environment is divided into one simulation runtime and multiple agent runtimes as shown in Figure 2 (a). Each runtime runs on its own Java virtual machine. The graph of the final bid prices for each number of agents is shown in Figure 3.

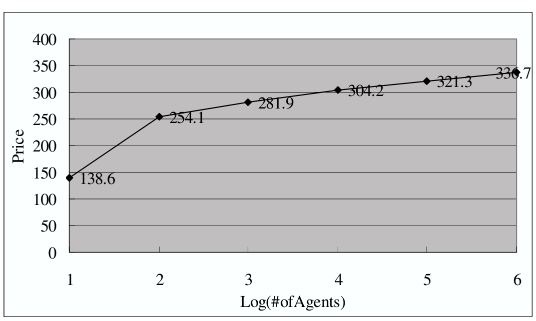


Figure 3. Final bid prices

Figure 4 shows the distributions of the final bid prices. We can see that the final bid price increases as the number of agents increases and its incremental ratio gradually shrinks as the number of agents increases. The distribution of the final bid price becomes more sharply peaked as the number of agents bidding increases.

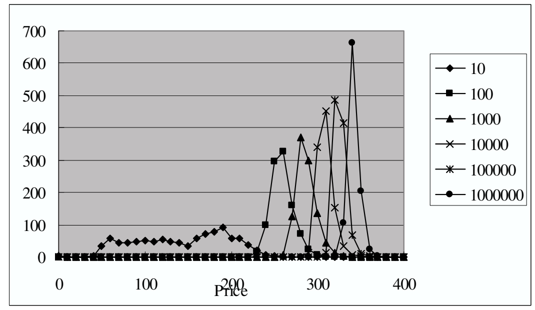


Figure 4. Distribution of the final bid prices

4. Performance Evaluation

The performance of the simulations is significant for an MABS. However, there is no consensus on performance metrics for MABS. Here, we use “aps” (agents per second) which is the total number of times of that agents handle messages during the execution of a simulation divided by the total simulation time. We measured the performance of the auction simulations for two system configurations: a symmetric multi-processor (SMP) machine and a system with multiple computers.

4.1 Performance on a SMP machine

To maximize performance, we divided the auction simulation environment into a simulation runtime and four agent runtimes except when the number of agents was only ten. For ten agents, we divided the environment into a simulation runtime and an agent runtime. The hardware and software configurations appear in Table 1. The results are shown in Figure 5.

Table 1. Configuration of the SMP environment

CPU	Dual Core AMD Opteron ™ Processor 275 2.19 GHz, 2-way
Memory	3.25 GB
OS	Microsoft Windows XP Professional Version 2002 SP2
Java	J2RE 1.4.2 IBM Windows 32 build cn142-20060421 (SR5)

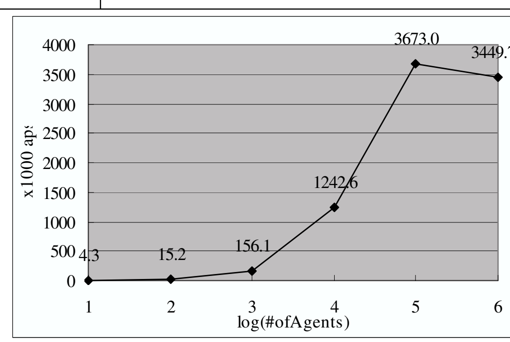


Figure 5. Performance on the SMP environment

The performance for 100 agents is 15244 aps and peaks at 3,670,000 aps for 100,000 agents. The reason why performance is small when the number of agents is small is that the processing overhead for an agent processing a message is much smaller than the processing cost of the other processes such as message transfer.

4.2 Performance on multiple computers

In this section, we show the performance measurement results where the auction simulation is executed on multiple computers. The simulation environment is divided as described in the previous section. The configurations of hardware and software are shown in Table 2. All of the computers have the same configuration. The number of agents was set as one million. The number of computers varied from one to six. The measured results are shown in Figure 6.

Table 2. Configuration of each computer in the multiple-computer environment

CPU	Intel(R) Pentium(R) 4 CPU 3.00 GHz
Memory	2.0 GB
OS	Linux version 2.6.9-34.EL
Java	J2RE 1.4.2 IBM build cxia32142-20060824 (SR6)
Network	Gigabit Ethernet

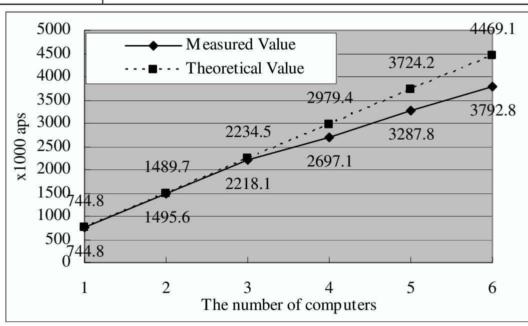


Figure 6. Scalability on the multiple-computer environment

The performance increases as the number of computers increases. For six computers, the performance is 85% of the theoretical limit (so the scalability ratio is 85%). We can say the system is scaleable up to at least six computers. However, performance is lower than the theoretical limit because there is overhead for the execution of the simulation runtime and for messaging. In addition, the time when an agent runtime finishes processing in a simulation cycle is not the same as when the other agent runtimes finish. Therefore, the total performance is somewhat lower than the theoretical limit.

5. Related Work

Many ABS simulation frameworks have been developed [2, 5, 6]. Recent research trends along with practical applications for large scale simulations of the real economy and complex networks with power law distributions call for the development of new frameworks. For example, MASON provides a toolkit to execute high speed simulations with many agents in a single process [3]. SOARS provides a common platform for education and realworld applications. It can utilize Grid computing [4]. On the other hand, ZASE aims at performing even larger-scale simulations by managing many agents in each process and by combining the processes. Takahashi proposed an agent-based simulation framework on the supercomputer BlueGene/L [7]. ZASE is a framework intended to construct a large-scale multiple-agent simulation system which can be executed on more general computing systems.

6. Conclusion

We introduced ZASE, is a platform for MABS. To demonstrate the use of ZASE, we presented agent-based auction simulations where the number of agents varied from ten to a million. The results indicate that the number of agents affects the final bid prices and their distributions. Performance measurement results on both an SMP computer environment and a multiple-computer environment were also shown. The results were about 3.5 millions aps in the both cases. For the agent-based auction simulation, the scalability ratio on the multiple-computer environment is 85% when six computers are used. The scalability ratio is such good value for the simulations because ZASE has low costs for message transfers among runtimes, but we need more investigations into the reasons for the performance degradations in the multiple-computer environment.

ACKNOWLEDGEMENT

This work was supported by "Strategic Information and Communications R&D Promotion Programme (SCOPE)" of the Ministry of Internal Affairs and Communications, Japan.

REFERENCES

[1] Repast, http://repast.sourceforge.net/

[2] Minar, N., Burkhart, R., Langton, C., and Askenazi, M., The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations, http://www.santafe.edu/projects/swarm/overview.ps , 1996

[3] MASON, http://cs.gmu.edu/~eclab/projects/mason/

[4] Deguchi, H., Tanuma, H. and Shimizu, T. , SOARS: Spot Oriented Agent Role Simulator - Design and Agent Based Dynamical System, Proceedings of the Third International Workshop on Agent-based Approaches in Economic and Social Complex Systems, pp.49-56. 2004

[5] Parker, M. T., What is Ascape and why should you care?, Journal of Artificial Societies and Social Simulation 4(1), http://www.soc.surrey.ac. uk/JASSS/4/1/5.html, 2001

[6] StarLogo, http://education.mit.edu/starlogo/

[7] Takahashi, T. and Mizuta, H., Efficient Agent-based Simulation Framework for Multi-Node Supercomputers, The proceedings of the 2006 Winter Simulation Conference, 2006 (to be shown)

[8] Mizuta, H. and Steiglitz, K., Agent-Based Simulation of Dynamic On-Line Auctions, Proceedings of the 2000 Winter Simulation Conference vol2, pp.1772-1777, 2000

[9] Yamamoto, G. Agent Server Technology for Managing Millions of Agents, International Workshop on Massively Multi-Agent Systems 2004, Springer pp.1–12, 2000