Architecture of High-Performance Fuzzy Multi-Agent System

Ivan Parasiuk, Sergiy Yershov

Glushkov Institute of Cybernetic of NAS of Ukraine, 40 Glushkov ave., Kyiv, Ukraine

iv.para@yandex.ru, sershv@ukr.net

Abstract. *Multi-agent simulation framework, which followed fuzzy type-2 fuzzy agent model and speeded-up the agent-execution and agent-messaging using computational power of GPUs in high-performance cluster environment, is developed. Architectural models of intelligent agents and multi-agent systems based on type-2 fuzzy logic, which enable to present more informative the degree of uncertainty of fuzzy rules in the specification of the behavior of these agents and systems. Approach for construction of fuzzy multi-agent systems in high-performance environment based on the specification of fuzzy graph transformation and adaptation of fuzzy rules by coordinator are explored. To reduce the cost of communication between agents we propose a method that is based on the generation of virtual blocks of agents on memory hierarchy of cluster system.*

Keywords

Multi-agent systems, fuzzy agents, cluster system, MPI, CUDA .

1 Introduction

The main concept of a multi-agent system is to simulate real world environments and interactions, composed by many entities, e.g. a building full of people during an emergency evacuation, market simulation, biological interactions between cells or enzymes, and so on. The biggest problem is that a multi-agent simulation leads to a huge amount of computing data, and it is hard to make it real time.

The GPU offers some enormous performance advantages for agent implementation over more traditional CPU-based alternatives. Using a hybrid cluster systems (GPU&CPU clusters), such as SKIT-4 [1], can further increase the complexity of multi-agent models and simulate systems of such complexity that meet real life situations. Intelligent fuzzy agents are greatly complicated by the need to consider not only functional requirements, based on a parallel algorithms for fuzzy decision-making, but features and details of specific paradigms of data and threads mapping onto heterogeneous systems that uses both Message Passing Interface (MPI) as well as Compute Unified Device Architecture (CUDA). To overcome these problems we developed model-based architectural approach to development of fuzzy multi-agent systems that operate within current and emerging cluster environments, including SKIT-4.

2 Architecture of Fuzzy Multi-Agent System

Architecture of fuzzy multi-agent system that consists of N motivated agents is shown in Fig. 1. All agents receive the same stimuli $\tilde{x} = \{\tilde{x}_1, ..., \tilde{x}_m\}$. Each agent applies the fuzzy graph rewriting rules and makes decisions about acceptable behavior $\beta_j = \{\beta_{j1}, ..., \beta_{jl}\}$. Production of final decision is carried out by the coordinator that also has a task of fuzzy rules adaptation. For this purpose, module of evaluation proposals from agents based on $\beta_1, \beta_2, ..., \beta_N$ and a set of motivations performs adjustment of rewriting rules for each agent and passes the result to module for result calculation. As a result of comparisons using fuzzy ranking index, the final decision of multi-agent system, which means "best" behavior of the proposed $\beta_i = \{\beta_{i1}, ..., \beta_{il}\}$, is produced [2, 3].

In coordinator of multi-agent system, original settings of motivation are used on the basis of fuzzy rules to determine the suitability of fuzzy agent behavior in different environments. To enable such adjustment, user selects values of motivations m_1, ..., m_N, to be used when calculating the fuzzy utility (that is motivation to help choose the best agent in a population) and one of allowable learning setting for the agent.

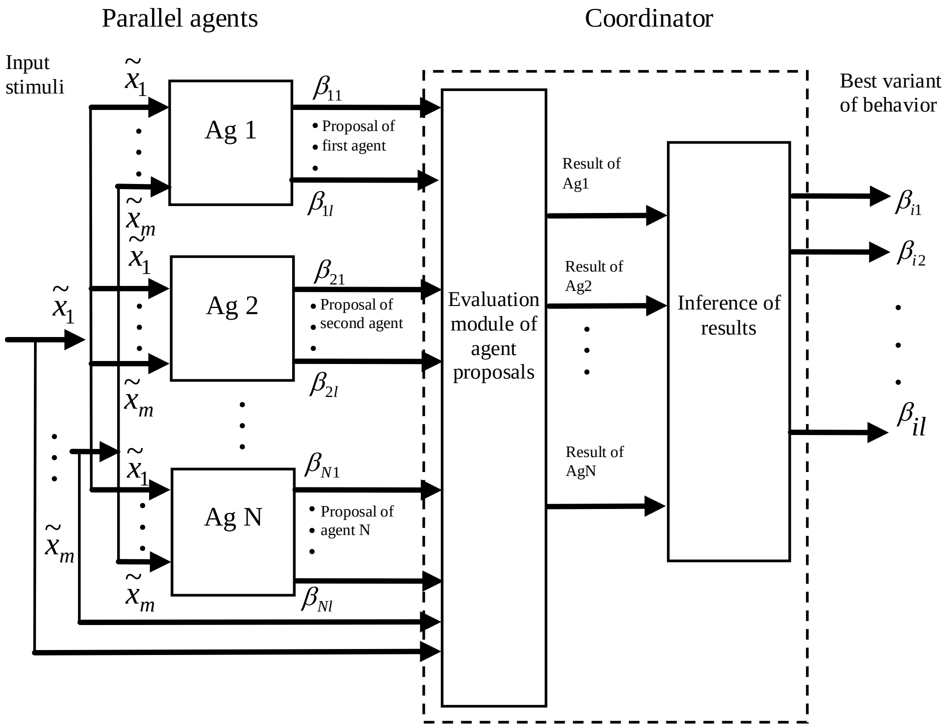

Fig. 1. Structure of fuzzy multi-agent system with coordinator

Module of evaluation of proposals performs following steps: based on genetic algorithm, it generates different populations of motivated agents; each agent population, in turn, produces and executes a consistent set of actions β_j; on the basis of the proposed behavior of each agent, the coordinator produces a set of N utility values; resulting fuzzy utility F is calculated on the basis of motives m_1, ..., m_N and individual utility values $f_1,..., f_N$. Evaluation finishes if maximum number of iterations of the genetic algorithm is reached. The best agent in the population remains as the final version of the agent.

We constructed several models of intelligent agents and multi-agent systems based on type-2 fuzzy logic, that enables more informative introduce uncertainty of fuzzy rules in the specification of the behavior of agents and systems.

Operational basis of intelligent agent that is based on type-2 fuzzy logic, is a system of **IF-THEN** fuzzy inference rules, for which sets of antecedents or consequents of rules are type-2 fuzzy sets (fuzzy numbers). Main components of architectural model of the fuzzy agent is fuzzyfier, system of rules, fuzzy inference means and processor of output values. The processor of output values consists of type reductor that creates the output fuzzy sets of type 1, and defuzzyfier that generates corresponding crisp value.

If assume that all is M rules for agent with p inputs $x_1 \in X_1$, ... , $x_p \in X_p$ and one output $y \in Y$, then i-th rule of agent can be represented as follows:

$$R^i: \quad \textbf{IF } x_1 = \tilde{F}_1^i \textbf{ AND} \dots \textbf{AND } x_p = \tilde{F}_p^i \quad \textbf{THEN } y = \tilde{G}^i, \quad i = 1,...,M. \tag{1}$$

Using in developing the agent of interval type 2 fuzzy sets and multiplication as t-norm, the result of intersection of all antecedent sets $F^i(x) = \prod_{k=1}^{p} \mu_{\tilde{F}_k^i}(x_i)$ is resulting in triggering interval, which is represented as a pair of values: $F^i(x) = \left[\breve{f}^i, \hat{f}^i\right]$, де $\breve{f}^i = \breve{\mu}_{\tilde{F}_1^i}(x_1) * ... * \breve{\mu}_{\tilde{F}_p^i}(x_p)$ – lower boundary of membership and $\hat{f}^i = \hat{\mu}_{\tilde{F}_1^i}(x_1) * ... * \hat{\mu}_{\tilde{F}_p^i}(x_p)$ – its upper boundary. The component "Type reductor" create fuzzy sets of type 1, called centroid, which is further converted into crisp values by defuzzyfier. Construction of the centroid of type-2 fuzzy sets is simplified if secondary membership functions are interval. Construction of the centroid $[\, y_l(x), y_r(x)]$ is reduced to minimize of triggering levels f^l of value $\min\left[\sum_{i=1}^{M} y_l^i f^i \Big/ \sum_{i=1}^{M} f^i\right]$ with constraints $f^i \in [\breve{f}^i, \hat{f}^i]$ and finding maximum value $\max\left[\sum_{i=1}^{M} y_r^i f^i \Big/ \sum_{i=1}^{M} f^i\right]$ where $f^i \in [\breve{f}^i, \hat{f}^i]$.

Purpose of "Defuzzyfier” component: based on aforementioned values to determine a single value that is result of application to input $x = (x_1,...,x_p)$ of fuzzy rules system (1): $y(x) = (y_l + y_r)/2$.

A multi-agent system architecture to achieve following aspects of the behavior of agents as maintaining distance, velocity coordination and bypassing of obstacles is developed, which are specified by means of type-2 fuzzy logic. Behavior of group consisting of *n* agents, in *m*-dimensional space can be defined by the following equations of motion:

$$p_i(t) = f\left(v_i(t), p_i(t-1)\right), \quad v_i(t) = g\left(u_i(t), v_i(t-1)\right), \tag{3}$$

where $p_i(t), v_i(t) \in \mathbb{R}^m$ – position and velocity vector of the agent, respectively, and $u_i(t) \in \mathbb{R}^m$ – influence vector for agent *i*.

To simulate "keep together" behavior are developed three fuzzy rules systems (to align intervals, to align velocity and avoiding obstacles) that correspond to each composite behavior. Each fuzzy rules system (FRS) has crisp input and then produces control effects $u^1{}_i$, $u^2{}_i$ and $u^3{}_i$. To determine the movement of each agent in the next step u_i in (3) is used fuzzy rules system of second layer, that receiving input vector of control effects, including $\tilde{u}_i$, given by expression $\tilde{u}_i = -c_1(p_i - \tilde{p}) - c_2(v_i - \tilde{v})$, where c_1 and c_2 are feedback factors. To evaluate a degree to which the behavior of agents meets the requirement of supporting interval, following function for a group of agents is introduced:

$$P(p) = \sum_{i}^{n} \sum_{j=1}^{n} \psi(\left| p_i - p_j \right| - d), \tag{4}$$

where $\psi(x) = x^2$, $\left| p_i - p_j \right|$ – module of difference of positions of agents *i* and *j* in space.

Using type-2 agents perform bypass of obstacles more smoothly than using rules of type 1, with lower standard deviation from the center of group [4]. Regarding function for maintenance of interval (4), its value for type-2 FRS is less than for FRS of type 1 on average. Moreover, the value of degree of deviation of agents velocities that are conducted by three type-2 FRS is less than for its corresponding conventional fuzzy rules.

3 Architectural Framework for High-Performance Multi-Agent Simulation

With the increasing number of values (variables) that are processed simultaneously by fuzzy agent, fuzzy rules system is much more complicated. The main reason for such complexity is the number of fuzzy rules, which is exponentially dependent on the number of input values and the corresponding linguistic terms (so-called exponential growth problem [5]). As a result of the increase in the number of rules, not only it takes more time to calculate the output values, but also hampered the understanding of the system of rules, which is especially important for systems based on fuzzy logic of higher order. One of approaches used to simplify the system of fuzzy rules is to construct an equivalent hierarchical fuzzy system [5]. Such a system may have subordinate structure in terms of fuzzy subsystems and relations between them. In this case, each subsystem is a base of rules given by (1), while each interaction is represented by an intermediate variable that links the pair of adjacent rule bases. The value of the intermediate variable coincides with the value of the output variable for the rule base of subordinated agent and the value of input variable of coordinator agent.

Generalization of this model of multi-agent systems is its representation as a virtual grid. Each agent is represented by grid node whereas an interaction between agents – by connections between nodes. A model of multi-agent systems with $p \times q$ nodes $\{N_{11}\ldots N_{p1}\},\ldots,\{N_{1q}\ldots N_{pq}\}$, $p \times q$ inputs of nodes $\{x_{11}\ldots x_{p1}\},\ldots,\{x_{1q}\ldots x_{pq}\}$, that acquire linguistic values for any valid input sets, $p \times q$ outputs of nodes $\{y_{11}\ldots y_{p1}\},\ldots,\{y_{1q}\ldots y_{pq}\}$, that acquire linguistic values of any admissible set of outputs, that consists of p horizontal levels and q vertical layers can be described by structure (5):

$$\begin{array}{ll} & \text{Layer } 1 \ \ldots\ldots\ldots \ \text{Layer } q \\ \text{Level } 1 & N_{11}(x_{11}, y_{11})\ldots\ldots\ldots\ldots N_{1q}(x_{1q}, y_{1q}) \\ \ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \text{Level } p & N_{p1}(x_{p1}, y_{p1})\ldots\ldots\ldots\ldots N_{pq}(x_{pq}, y_{pq}) \end{array} \tag{5}$$

Grid structure in (5) determines location of agents and their inputs and outputs. In this case, each input and output can be both scalar and vector representing a set of type-2 fuzzy values. Levels in the grid structure are dimension hierarchies of nodes in terms of subordination in space, whereas the layers define time dependence in terms of execution sequence. For uniformity in multi-agent systems that are described by (5), it is suggested that there are nodes in each cell of the basic grid structure. However, not all cells in the model are required to be agents. For simplicity the connections in the above model are between nodes of the same level and nodes in adjacent layers. However, some connections in multi-agent systems can be among agents at different levels or between agents at non-adjacent layers.

As a target platform for implementation of multi-agent systems, cluster system SKIT [1] is used. To support interaction between intelligent agents a special library that contains a number of patterns, classes, and methods to conduct of agents behavior (i.e cyclic, single and repeated) is created. The library is based on exchange of different types of asynchronous messages using MPI [6]. Variant of the library is implemented for use of the software and hardware stack of CUDA [7]. The library enables to organize the system model as a virtual grid of agents, in which agents are allowed to interact only within a limited distance (number of transitions on the grid). Because of distribution of global state between processors, part of state of individual agents may persist in the neighboring agents. In the cyclic execution of fuzzy inference algorithms, elements of this state can be requested before start the next iteration. Special barrier primitives are used to synchronize execution of iterations (stages of execution) of fuzzy agents. However, it does not take into account the variety of memory types and latency of message transfers between agents that use these types of memory. Therefore, to reduce the cost of communication we propose a method that is based on a virtual grid hierarchy on memory types of agents.

Virtual network of agents is partitioned into blocks of agents assigned to the individual processing elements (CPU, GPU). Each block consists of $L \times L$ agents, which are surrounded by E layers, which are agents of the neighboring blocks. This calculation within the local block may be extended to E iterations, then it becomes necessary to transfer data between blocks. Since the block of size $L + 2E$ contains all the information required for local computation, this leads to fewer exchanges in the grid. The cost of communication after E iterations can be represented in different ways: for CPU/GPU respectively $C_{CPU} = c_w(L^2 - (L - 2E)^2) + c_r((L + 2R)^2 - L^2)$, $C_{GPU} = c_w(L^2) + c_r((L + 2R)^2 - L^2)$, where c_w, c_r – time spent to read (write) data one agent on average.

In this framework, at lowest level of the hierarchy are agents that run as CUDA threads. Even at this level, reducing of cost communication is achieved through the use of agent blocks $L + 2E$ as CUDA blocks. Higher blocks are agents at some GPU, synchronization among which is due sequential execution of CUDA kernels. At highest level, largest blocks of agents are distributed among nodes, communication among which is based on MPI. After E_{GPU} iterations for agent blocks of compatible ranks are made non-blocking calls MPI_Irecvs and MPI_Isends, that update states of $(L + 2R)^2 - L^2$ agents, and then its execution continues.

4 Conclusion

We developed multi-agent simulation architectural framework, which followed fuzzy type-2 fuzzy agent model and speeded-up the agent-execution and agent-messaging using computational power of GPUs in high-performance cluster environment. Architectural models of intelligent agents and multi-agent systems based on type-2 fuzzy logic, which enable more informative to present the degree of uncertainty of fuzzy rules in the specification of the behavior of these agents and systems. Approach for construction of fuzzy multi-agent systems in high-performance environment based on the specification of fuzzy graph transformation and adaptation of fuzzy rules by coordinator are explored. To reduce the cost of communication between agents we propose a method that is based on the generation of virtual blocks of agents on memory hierarchy of cluster system. Fuzzy agent’s communication through messages has been implemented with efficient use of GPU shared memory; a resulting speedup over the original CPU-based implementation of over 12000 times has been achieved.

References

[1] Supercomputer of Glushkov Institute of Cybernetics of NAS of Ukraine. – http://icybcluster.org.ua.
[2] Parasiuk I.M., Yershov S.V. Multi-agent systems based on fuzzy models: architecture, coordination and transformations // Bulletin of Kyiv National Taras Shevchenko University. Physical and mathematical sciences. – 2011. – № 1. – C. 179–186.
[3] Yershov S.V. Transformational approach to development of adaptive intelligent agents based on fuzzy transition schemes // Computer mathematics. – 2011. – № 1. – C. 69–78.
[4] Parasiuk I.M., Yershov S.V. Multi-agent model based on fuzzy logic of higher type for high performance environment // Programming problems. – 2012. – № 2–3. – C. 260–269.
[5] Wang D., Zeng X., Keane J. A survey of hierarchical fuzzy systems // International Journal of Computational Cognition. – 2006. – Vol. 4, № 1. – P. 18–29.
[6] Sloan D. High Performance Linux Clusters with OSCAR, Rocks, OpenMosix, and MPI. – 2005. – 350 p.
[7] Sanders J., Kandrot E. CUDA by Example: An Introduction to General-Purpose GPU Programming. – 2010. – Addison-Wesley Professional. – 312 p.