Placing Agents in Massively Multi-Agent Systems

Naoki MIYATA and Toru ISHIDA
Department of Social Informatics Kyoto University
miyata@lab7.kuis.kyoto-u.ac.jp, ishida@i.kyoto-u.ac.jp

1 Introduction

In this paper, we propose a technique which determines the placement of agents, taking both the computation load of agents and the interaction among agents in distributed multi-agent systems into account. We formulate the agent placement problem as the problem that the load of agents is distributed among multiple agent servers and the interaction cost arising among agent servers is minimized. Then, we provide an approximate algorithm to the agent placement problem by means of using an idea of an approximate algorithm for network partition problem. In this paper, we conduct a computer simulation to test the effect of our technique.

2 Agent Placement Problem

When agents are distributed among agent servers to distribute the load of agents, two points have to be considered. They are the computation load of each agent server and the interaction cost arising among agent servers. We formulate the agent placement problem, taking these points into account. $p(a_i, a_j)$ is the interaction frequency between an agent $a_i$ and $a_j$. $A'_i$ is the set of agents residing at an agent server $s_i$. $W'_i$ is the computation load of $s_i$.

$$\min \sum_{a_i \in A'_i \land a_j \in A'_j (k<l)} p(a_i, a_j)$$

$$s.t. W'_i < T_h \ (i = 1, 2, ..., m)$$

In this research, we use the idea of the k-way FM algorithm for an approximate algorithm to the agent placement problem. Our algorithm consists of two phases. The first phase is initial placing phase and the second phase is improve cost loop phase, which reduce the interaction cost of the initial placing.

3 Simulation of Placing Agents

In this simulation, we apply three techniques to a distributed multi-agent system and record the change of interaction cost in the simulation to compare them.
In this simulation, the number of agents $n$ is 10,000 and the number of agent servers $m$ is 50. The interaction frequency $p(a_i, a_j)$ is set to a random number. And after the simulation start, the load $w_i$ of each agent is updated by a random number every iteration. We apply each technique to the agent server, the load of which is larger than the threshold $T_h$, to distribute the load.

The simulation is conducted 10 times in each applied technique. The average of the interaction cost is plotted in the Fig.1. The vertical axis is the sum of the interaction cost among agent servers. The horizontal axis is the iteration time in the simulation. This graph shows the interaction cost in the simulation applying WBLB, Comet and our technique every iteration respectively. WBLB Algorithm places agents based on the load of the agents in order to distribute the computation load among agent servers. Comet Algorithm allocates agents based on the computation load and the interaction.

As represented in Fig.1, our technique decreases the interaction cost more effectively than the other algorithms.

4 Research Contribution

In this paper, we have established a technique to improve the performance of distributed multi-agent systems effectively. To establish this technique, we have formulated the agent placement problem based on both the load of agents and the interaction cost among agents, and provided an approximate algorithm to the formulated problem. Then, we carried out a simulation of applying our technique to the massive navigation system to observe the change of interaction cost. The main contribution of this paper is that we formulate the problem on distributing agents among agent servers and provide an approximate algorithm to the problem. We have verified that our technique can effectively decrease the interaction cost by a simulation.