International Journal of Engineering Innovation & Research
Volume 4, Issue 4, ISSN: 2277 – 5668

Optimization Cost of Agent's Migration for Load Balancing in Distributed Systems

Maryam Jafari
Applied Sciences University Municipality of Andimeshk, Andimeshk, Iran
Email: maryamjafari_rk@yahoo.com

Hossein Shirazi
Department of Information Technology,
Malek Ashtar University of Technology, Tehran, Iran
Email: shirazi@mut.ac.ir

***Abstract* – One of the most important factors in distributed multi-agent systems is load balancing that can improve the performance of these systems effectively. In the existing methods scheduling optimization was done for load balancing. However, the migration co optimization and security, has not been investigated. Therefore the method that has the best performance in minimum time, and reduces the cost of migration is needed. Load balancing Algorithm in distributed multi-agent systems in terms of migration cost of agents is studied. The proposed algorithm by reducing load balancing algorithm runtime tried to reduce the communication and migration cost. The simulations were carried out to investigate the performance of the algorithm show that the algorithm has less time for scheduling in distributed multi-agent systems for load balancing and reduce the migration cost.**

***Keywords* – Agent, Migration, Communication, Distributed Systems.**

I. Introduction

In multi-agent systems, the "agent" has an essential role. Thus defined, the definition of multi-agent system operating discussed. There is no single definition for the agent since now[4]. Agents live in same environment. They can sense their local environment, and interact with other agents in their environment. Agents attempt to achieve specific goals or specific tasks. They respond to changes that may occur inside them. They create a community to collaborate to achieve their goal and common goal [11].In general, the behavior of an agent forms based on the objectives and operations of the system. Targets of System can be in line with system global optimality, system management and system control. The Agent has The basic operating characteristics such as purpose, knowledge, labels, etc., That are formed of functional units, communication model, business process model, inference model, study model, data transmission, etc.,., [8].

Multi-agent system has been formed of autonomous agents that dynamically cooperate and coordinate to achieve the objectives of national and local purposes. The main advantage of multi-agent systems is that agents do not rely on the special existence to execute their orders. Failure of one component does not interrupt whole system operation [17].

Dynamic load balancing is allocation of limited resources among multiple tasks over time [12]. The multi-agent systems have limited resources and time constraints, therefore, needed to focus on agent schedules in them. There are problems in dynamic load balancing of distributed multi-agent system. In the concentrated way there are two problems: the bottleneck and failure point. Due to heuristic nature of the partitioning method, it is much time consuming. The method of the immigration should be considered. Once proper purpose Search and didn't find, search must be done again. By dividing societies, the cost will increase [12].The Goals of the load balancing algorithm is to minimize time and cost in distributed multi-agent systems.

The proposed algorithm by reducing load balancing algorithm runtime tried to reduce the communication and migration cost.

We investigated load balancing algorithm in distributed multi-agent system for reducing the cost of communication and agent migration. In the proposed algorithm [9] is tried to reduce the time and cost of load balancing. However, the migration cost is not studied.

This paper introduces load balancing algorithm for distributed multi-agent systems using all system resources to reduce algorithm execution time and cost of agent communications and agent migration cost.

In previous methods, the migration cost is not studied. The proposed algorithm performs distribution of computation load among servers for improvement of system performance. And reducing operating costs of migration have been investigated.

The proposed algorithm achieves optimized scheduling using parallel computing on all servers and partial information of agent communication. The way was adopted for migration Agents selection and destination server selection that reduce communication costs and agent migration cost.

In this paper, influence of two factors reducing communication costs between agents and reducing the number of immigration agents to select the destination server has been investigated.

II. Related Work

In this section, we brought related work on dynamic load balancing in distributed multi-agent system.

A. Load balancing in multi-agent system

Balanced Dynamic load is allocation of limited resources over time among multiple tasks[12]. According to the above definition, we are faced with limited resources and time. In load balancing it is assumed that the right amount of work would be dedicated to each processor in order to increase performance and to reduce the run time of programs. therefore, Distribution of a heavy work between nodes needs load balancing. In many cases, load balancing are required such as parallel applications [1] which right use of processing capacity should be assured. For applications that condenses large amount of data processing and computing needs.

Dynamic load balancing is allocation of limited resources among multiple tasks over time [12]. The multi-agent systems have limited resources and time constraints, therefore, needed to focus on agent schedules in them. There are different methods for load balancing such as partitioning, based on the economic theory, rule-based approach, and adaptive techniques. Partitioning method can properly distribute the workload generated by agents between the available distributed resources [12]. For example, adaptive method [6], fast adaptive balance method [18], considering the irregular shape of the regions [16]. However the partitioning methods have heuristic nature, and spend much time, so they significantly are affected by the average response time [12]. Paper [5] proposes load balancing model based on given credit to each agent and it improve the volume loading more than balance oriented normal programs. Jang and Agha, 2004 proposed two mechanisms to improve the adaptive agent allocation: one mechanism aims minimizing agent communication cost, while the other mechanism attempts to prevent overloaded computer nodes from negatively affecting overall performance. In the paper [13] network of agents considered as a "small world", so the load balancing is performed based on the community. This method evaluates the communities to select the most appropriate set of agents to be moved. Shin *et al.*, 2009 proposed, load balancing based on the agent mode that it performs efficient and justly allocation of resources to agents. This method spends much time for searching the best mode.

Another scheduling algorithm is the distributed approximate optimized scheduling algorithm with partial information (DAOSAPI) [12] this algorithm needs partial information of agent communication for searching, and also don't need agent status information.

In the paper [9]optimizing the scheduling algorithm of load balancing in distributed multi-agent systems has been made. But reducing communication cost and agents migration have not been studied. In this paper we use the algorithm presented in paper [9] to reduce communication cost and migration cost.

B. Agent distributed and Agent communication architecture

Due to architecture of distributed multi-agent systems, as the nodes are distributed in the environment. Each node has some agent to do tasks and according to the position of the nodes, the tasks given to it. There are communication, cooperation and coordination between agents. In a distributed multi-agent system communication model between the agents is frequently changing. On a large scale multi-agent distributed system this problem can cause trouble. When agents are distributed in multi servers spending interplay cost between agents in the load balancing to improve the efficiency of the system should be considered.

The goal of load balancing is to minimize response time and cost. When one of the nodes is given more tasks than its ability, load imbalance would be created in it. In this case, the response time is increased in that one. Not possible to avoid this situation-because the costs of the ongoing investigation before delivering task will increase. So a solution to restore balance in the nods is needed. To solve the problem, a self-load balancing infrastructure that responds quickly to changes in the environment could be used. Creating a Self-requisite infrastructure require the mobility of agents. Dynamic load balancing is a possible application of agent mobility [2].In fact, mobility is a characteristic of the variety of multi-agent system which is in weak or strong form.

Most of the operating systems that support the mobility of agents are weak mobility. There are a limited number of operating system that support strong agent's mobility such as JIAC [7]، NOMADS [15] and Organic Grid [3].

III. Analysis of Dynamic Load Balancing in Distributed Multi-Agent System

Assume multi-agent systems have M server that n agents are among those distributed. The load capacity of all servers is the same that divided among agents resident in server equally. Agents within the same server and two different servers can communicate with each other for cooperation and coordination.

The agents communication cost within the same server (intra-server interaction) is much less than two different servers communication cost. Four objectives that followed to improve the system performance are:

- Agent load distribution among servers
- Reducing the communication cost between servers
- Reducing load balancing algorithms execution time
- Reducing migration costs of agents

Due to reducing connection cost between agents in similar server compared to different servers, in load balancing algorithms tried agents linked together are placed on the same server. Reducing the cost of agent migration, instead of specific agent migration, set of agents can be migrated and the number of destination servers for migration will also be reduced. So server will be selected that reduces operating costs between agents and has more capacity in accepting largest number of immigration agents. To reduce the running time of load balancing algorithm parallel processing used.

IV. Proposed Load Balancing Algorithm in Distributed Multi-Agent System

In this section we define variables used for extra load, the number of migration agents and number of migrations. Then the algorithm rules apply.

Used variables in the algorithm described in the paper [9] have been used.

International Journal of Engineering Innovation & Research
Volume 4, Issue 4, ISSN: 2277 – 5668

OL_{Sj} Shows overloading in server s_j, which is calculated by the following equation.

$$OL_{Sj} = L_{Sj} - Th \quad (1)$$

nm_{Sj} Displays the number of immigration agents can be obtained by the following equation.

$$am_{Sj} = \frac{OL_{Sj}}{La_{ij}} \quad (2)$$

The amount of agent migration number nm_{Sj}, can be calculated by selected destination server's capacities.

Table I: Notations [9]

s_j	Server j (j = 1,2,….,m)
a_{ij}	Agent i in server j ($i = 1,2,3,\ldots, n_j = \sum i$, $N = \sum_{j=1}^{M} n_j$)
L_{s_j}	computation load of s_j
$L_{a_{ij}}$	computation load of a_{ij}
L_{ps_j}	computation load acceptance capability of server s_j
$C_{a_{n_i}a_{n_j}}$	communication count between a_{n_i} and a_{n_j} ($a_{n_i} \in S_i$, $a_{n_j} \in S_j$, i,j ≤ M)
$C_{s_o s_j}$	communication count between s_o and s_j (o,j ≤ M ^ o≠j)
$C_{a_{io} s_j}$	communication count between a_{io} and s_j (o,j ≤ M ^ o≠j)
$C_{m_{oj} s_j}$	communication count between m_{oj} and s_j (o,j ≤ M ^ o≠j)
$C_{m_{oj} s_o - m_{oj}}$	communication count between m_{oj} and $s_o - m_{oj}$ (o,j ≤ M ^ o≠j)
Th	Threshold of computation load of servers

In this work we try to reduce number of immigration agents and thus achieve cost reduction in agent's migration. With the migration of selected agent set to destination server immigration expenses is reduced. However, due to the consideration of reducing communication cost in choosing the destination server and sets of optimal agent migration, the number of target servers, and so migration agent increases. Therefore, the migration cost and time will also be increased.

In this paper, influence of number of factors the choice of moving to a server in the server as the destination server will show that the number of immigration and migration cost and reduced execution time.

Certification of selected destination servers is calculated by the following equation:

$$\text{Max}\,(am_{Sj}) \quad (3)$$

To reduce the communication cost, Collection of agents that have the most communication with destination server and the least communication with overload server is selected for the migration to destination server. Using the following equation, the optimal set agent is selected to migrate to the server Sj:

$$\text{Max}\ (C_{moj,sj} - C_{moj,so-moj}) \quad (4)$$

$C_{moj,sj}$ is server connection value Sj, with the optimal agent set moj selected for that. Communication value of server So with optimal migration agent set to Sj server is $C_{moj,sj} - C_{moj,so-moj}$.

V. Optimization Cost of Agent's Migration for Load Balancing

In this paper, the validity of the following equation calculates servers are:

Max (am_{Sj})

If $am_{Sj} >= am_{So}$, means that Sj server is accepting all migration agents of So server. So migration agent set is migrated in groups to destination server by server So. otherwise the MA in So broadcast the message contain (ID of So, number of rest migration agent , the agent load and new partial information about agent communication) and the algorithm repeats of step 3untilallloadistransferred. At end The MA in So begin immigration procedure.

Agent migration process is shown in the following figure. After ordination of manager agents in over load and destination server, the selected optimal agent set for that server is migrated. Consequently when the number of execution of balancing algorithm reduces, time and cost of agent migration will be reduced.

In this paper, reducing the cost of migration and run time for load balancing and decreasing the cost of communication has been studied. Using the agent's group migration instead of alone agent, costs of migration dramatically decreased. Instead, if reducing communication cost for agent migration to be considered in, as a result migration factor increases and will increase the cost of migration. In this paper, influence of number of factors in choosing destination server was investigated. Considering the number of immigration agents to selected destination server, a server can be chosen so that decreases communication and immigration costs of agents by reducing number of destination servers.

VI. Analysis of Time Complexity of the Proposed Algorithm

Considering the number of factors in the migration destination servers election algorithm time complexity is not changed. Time complexity as proposed in [9] is.

The proposed algorithm has two important pieces of code that a piece of code is implemented in over loaded server and other piece of code is implemented on other servers at same time. The piece of code on overloaded server is selection of the destination server for migration agents and the time complexity of it is O (M). The second piece of code is implemented on other servers for selects optimal agent set. The time complexity of second piece is O (NC). (N is the average number agents per server and C is the average number communication per agent). The time complexity of the proposed algorithm is O (MNC). Although the proposed algorithm has a low communication delay, it has no significant impact on its performance.

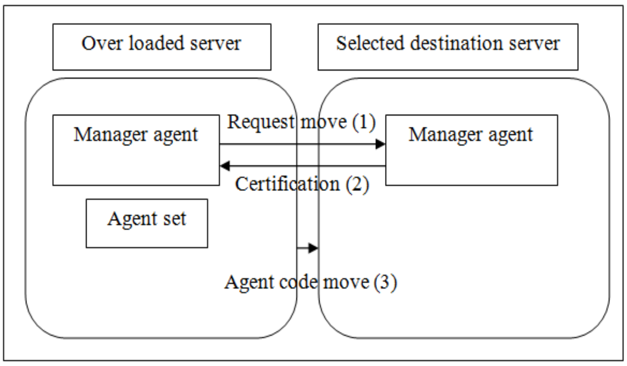

Fig.1. Agent's migration process

VII. Experimental Results

This section describes simulation to test our technique. Results obtained from the proposed algorithm compared with paper [9] and SLB. This simulation was implemented in C# and executed on a machine with 2.67GHz CPU and 4GB of memory.

A distributed network, which is the number of servers,is10,considered. The number of agents per each server has beenconsidered100, 200, 500, 1000and 1500whichis connected randomly other agents within the same server and other servers. Load Distribution among servers is randomized. The Computation threshold of servers is equal Average computational load of all servers. When a server overloaded, load balancing algorithm will run, and optimal agent sets for migration and destination server with best condition will be selected.

A. Simulation Results Analysis

According to the results, reducing the number of migration agents cause to reduce the cost of agents' migration and algorithm time. Reducing the number of algorithm execution for migration of all agents, the number of migration agents set, cost of agents' migration and algorithm running time reduces.

Table II: Comparison of execution time between SLB, DAOSAPI, Paper [9] and Proposed algorithm versus the number of agents per server

Number of agent per server	100	200	500	1000
SLB (s)	0.01	0.03	0.11	0.42
DAOSAPI (s)	0.031	0.047	0.062	0.078
Paper[ix]	0.01	0.015	0.031	0.046
Proposed (s)	0.01	0.015	0.02	0.03

This algorithm selects agents of the overloaded that have the most association with the destination server and the least association with the overload server. Hence reducing communication cost.

The proposed algorithm due to evaluation, selection and migration of set of agents instead of a single agent, reduces less communication cost compared to SLB and paper [9]algorithms.

Table III: Comparison of number of algorithm execution for migration of all agents between Paper [9] and proposed algorithm versus the number of agents per server

Number of agent per server	100	200	500	1000	1500
Paper[ix](s)	2	2	3	3	4
Proposed (s)	1	1	2	2	3

SLB algorithm evaluates all agents and among them migrate agents that had the highest external communication, and lowest internal communication to server that connected to the most. The result could reduce the communication cost much more than the proposed algorithm. But when compared to time that spend, and due to the frequent changes in the relationship between agents and system dynamics, is not Considerable amount.

To compare costs between the proposed algorithm, SLB and paper [9] a test has been conducted. Calculate the costs of collection interval equal to 100 seconds. The relationship between operating costs equal 0.1 KB / sec, has been considered. The results of the simulation are shown in Figure 2.

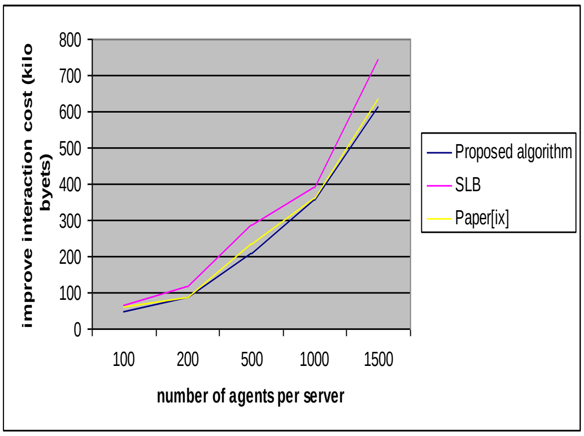

Fig.2. Comparison of reduces communication cost between SLB, Paper[9]and Proposed algorithm versus the number of agents per server

VIII. Conclusion

Simulation results of the proposed algorithm in comparison to [9] algorithm shows reduction in costs of immigration and agent sets. In contrast, it was found that increasing the number of agents in the system, the proposed load balancing algorithm execution time reduces more than other compared algorithms. The proposed algorithm achieved agents migration cost reduction by reduction in search time for the target servers and agent migration sets, and number algorithm execution.

References

[1] Banicescu I., Velusamy, V., (2002) 'Load Balancing Highly Irregular Computations with the Adaptive Factoring', Proceedings of the 16thInternational Parallel & Distributed Processing Symposium, IEEE, Los Alamitos, California,.

[2] Cabri, G., Leonardi, L., Zambonelli, F., (2000) 'Weak and strong mobility in mobile agent applications'. In Proceedings of the 2nd International Conference and Exhibition on the Practical Application of Java.

[3] Chakravarti, A., Baumgartner, G., Lauria, M., (2004) 'The Organic Grid: Self-Organizing Computation on a Peer-to-Peer Network'. In International Conference on Autonomic Computing (ICAC'04), New York, USA, May, pp. 96-103.

[4] Chen, J.J., Su, S.W.,(2003). Agent Gateway: a communication tool for multi-agent systems, Inf. Sci. Vol(150). pp.153–164.

[5] Chow, K.P., Kwo, Y.K., (2002) 'On load balancing for distributed multiagent computing', IEEE Transactions on Parallel and Distributed Systems, Vol(13), pp.787–801.

[6] Deng, Y.F., Peierls, R.F., Rivera, C., (2000) 'An adaptive load balancing method for parallel molecular dynamics simulations', Journal of Computational Physics, Vol(161), pp.250–263.

[7] Fricke, S., Bsufka, K., Keiser, J., Schmidt, T., Sesseler, R., Albayrak, S., (2001) 'Agent based telematic services and telecom applications'. In Communications of the ACM, Vol(44).

[8] Guo, Q, Zhang, M., (2010) 'An agent-oriented approach to resolve scheduling optimization in intelligent manufacturing', Robotics and Computer-Integrated Manufacturing Vol(26), pp.39–45.

[9] Jafari M, Shirazi H, RazaviEbrahimi S.A, (2013), Optimization Load Balancing Scheduling Algorithm in Distributed Multi-Agent Systems, International journal of computer Science & Network Solutions, Vol(1), pp. 32-44.

[10] Jang, M.W., Agha, G., (2004) 'Adaptive agent allocation for massively multi-agent applications', International Workshop on Massively Multi-Agent Systems, pp. 25-39.

[11] Kumar, R., Sharma, D., Sadu, A., (2011) 'A hybrid multi-agent based particles warm optimization algorithm for economic power dispatch', Electrical Power and Energy Systems,Vol(33), pp.115–123.

[12] Long, Q., Lin, J., Sun, Z., (2011) 'Agent scheduling model for adaptive dynamic load balancing in agent based distributed simulations', Simulation Modelling Practice and Theory 19, pp.1021–1034.

[13] Miyata, N., Ishida, T., (2007) 'Community-based load balancing for massively multi-agent systems', International Workshop on Coordination and Control in Massively Multi-Agent Systems, pp. 28-42.

[14] Shin, S.Y., Lee, H.C., Song, S.K., Youn, H.Y., (2009) 'A load balancing scheme for multi-agent systems based on agent state and load condition', International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 121–127.

[15] Suri, N., Bradshaw, J., Breedy, M.R., Groth, P.T., Hill, G.A., Jeffers, R., (2000) 'Strong mobility and fine-grained resource control in nomads'. In Proceedings of the Second International Symposium on Agent Systems and Applications and Fourth International Symposium on Mobile Agents, ASA/MA 2000, Zurich, Switzerland, volume 1882 / 2004 of Lecture Notes in Computer Science, Springer-Verlag, pp.2-15.

[16] Vigueras, G., Lozano, M., Orduna, J.M., Grimaldo, F., (2010) 'A comparative study of partitioning methods for crowd simulations', Applied Soft Computing, Vol(10), pp.225–235.

[17] Xianga, W., Leea, H.P., (2008) 'Ant colony intelligence in multi-agent dynamic manufacturing scheduling', Engineering Applications of Artificial Intelligence, Vol(21), pp.73–85.

[18] Zhang, D.L., Jiang, C.J., Li, S., (2009) 'A fast adaptive load balancing method for parallel particle-based simulations', Simulation Modeling Practice and Theory, Vol(17), pp.1032–1042.