A Method of Solving Message Chasing In Mobile Agent

Jinghua Wang[1], Yuanyuan Chen[2], Tingting Guan[3],Huan He[4]

huazhong normal university ,computer science, wuhan, china

chenyuanfc@163.com

Abstract: The technology of mobile Agent is an important method of computing in distributed system. Communication is a fundamental ability that enables mobile agents to cooperate with each other. However, due to the mobility and autonomy there still exists some problem in designing communication mechanism of reliability for mobile agents, such as communication failure and message chasing. A novel "Agent strategy of speeding limiting "method is proposed for message chasing. Simulation results show that proposed method provides promising solutions than the method of "Agent strategy of speeding waiting".

Keywords: mobile agent; message chasing; agent strategy of speeding limiting

1 Introduction

Mobile agent is a computer program, a moveable software, it can move from one host to another, it can also search the appropriate resources instead of the user to a number of problems for effective and reliable communication between mobile agents. The physical location change of agents will result in a problem of communication failure, that is, before a message gets to one host, the target agent has left away, making itself unable to receive this message[1]. The problem widely exists in mobile agent systems. Many methods also are proposed in other documents. In some extreme situations, like the agent moves frequently, whenever a message reaches a host where the target agent used to reside, the agent has just left away so it can not get the massage all the time and the message keeps chasing the recipient around the system but never gets delivered, resulting in a race condition which is called Message Chasing[1]. If the agent cannot receive messages sent to it in time, the collaboration will fail and the system may even crash. Therefore, in order to carrying out a reliable and efficient communication, the hot issue of research recently is how to solving the above problems.

For the convenience, we always assume that the network consists of fault-free FIFO (First In, First out) channels. That is to say messages will be transferred from one side of the channel to another orderly without any Transmission fault [1].

The remainder of this paper is organized as follows. Section 2 contains an analysis of the solving of the chasing problem. Section 3 contains the research of the method for prevent message chasing. Performance analysis in section 4, and finally conclude the summary of our work in section 5, and propose the further work in future.

2 Related works

The question of message chasing, that is, when the agent moves frequently, the velocity of delivering message can't reach the agent has just left away. In such case, message keeping chasing the recipient, but never reached, resulting in the message can't reach recipient forever.

The central server scheme refereed in the document [2], in some extent the scheme can solve the problem of message failure and message chasing. The central server scheme communication is synchronous, it can be forced to stop mobile agent. If there is news that it be delivered to an agent, then the agent must stop to wait for the delivered message. The target agent can move until the message reached it. For that, the mobility features of mobile agents have been limited.

In the document [3], a reliable communication mechanism for mobile agents named EMFS (efficiently message forwarding scheme) is also brought forth. The MEFS provides a location-transparent tracking mechanism and, in particular, guarantees message delivery under any condition in a fault-free network. At the same time, there also proposed an effective solution of message chasing problem which caused by the mobility of agent.

"Session-Oriented Communication" in document [4] implemented in Mole system uses the method of "request-and-reply" to establish communication between a pair of agents.

In document [5] take the forwarding pointer scheme tracking, its goal is to keep a forwarding pointer pointing to the next host on the path. Also there is a "home" of the

978-1-935068-01-3

target object. Messages are sent to home and then forwarded to the recipient along the path directed by the pointer. However, a racing condition may occur if the target agent moves frequently while document provides no solutions to it.

3 Ways to solve the message chasing problem

3.1 Introduction of communication model

In order to solving the message chasing, a mobile Agent communication model is proposed in Figure 1.

Agent is divided into multiple domains by the approach of the relationship between physical locations. We can see each region has a number of places. There is a "LS" module and "Home" module in every region, the two modules responsible for managing the whole address of mobile agent in every region. There is a module named mailbox, it is used to store messages which send to the agent in every domain. Also, there exists a module named "NS", it can administrate the whole address of mobile agent with all regions.

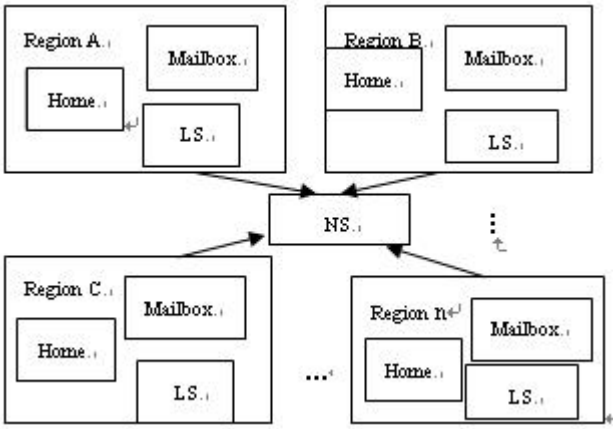

Figure 1 mobile agent communication model

The main data structure of the modules in the above picture with mobile agent communication model.

Home(AgentID,LSaddr,condition,tag):"AgentID" parameter represents the name of mobile agent in each region, "LSaddr" parameter represents the address of mobile agent, "condition " parameter represents a sign of whether moving or not, "tag" parameter represents the sign of message condition.

LS(AgentID,paddr,next,timeout):"AgentID" parameter represents the name of mobile agent in each region, "paddr" parameter represents the address of the current node, "next" parameter represents the address of the next region the mobile agent moved to, and "timeout" parameter represents the status of overtime.

Mailbox(id, destination name, mcount,count): "id" parameter represents the message number, "destination name" parameter represents the purpose of Agent's address, "mcount" represents the number of messages received, "count" represents the number of messages transmitted。

NS (AgentID, lsaddr): "lsaddr" parameter represents the address for each region.

Comparing to the document [3], two improvements of the model's main structure are mentioned as follow:

(1) Each node can temporary store and forward message through the mailbox.

(2) each agent has a variable named "number", it is used to record the number of node which the agent migrate, the initial value of the variable is 0, its value increases by 1 after passing a node.

3.2 The process of agent moving and the message forwarding

A complete communication mechanism of mobile agent has two parts at last: Agent tracking and message forwarding[5].

The common ways in agent tracking are central server scheme tracking、forwarding pointer scheme tracking、broadcast scheme and hierarchical scheme.

In the above ways, each has its own advantage and disadvantage. Such as, central server scheme more easily has the problem of bottlenecks under the condition that there are a large number of agents. Because all the agents compete for the central server. In the hierarchical scheme, there will lead to message chasing and communication failure. If a frequency of agent moving, the scheme also is not suit for the large-scale migration. As for the broadcast scheme, in the large application system of mobile agent, it will afford the high cost of communication, also the lack of availability, and can't guarantee the reliability of message forwarding. So in order to solving problems which proposed in our paper, we connect the central server scheme and forwarding scheme together in this paper.

According to the local region of communication and

cross-border communication, messages sending are divided into two situations: one is directly forward by looking after the news of the Agent in the local domain; the other is that it must search for the purpose of Agent's address in the NS module. Even if the agent moved in the process of message forwarding, but when the agent moved from one place to another place there also has a field in the LS module named “next”, the field stored the pointer address, so we can easily know the agent whereabouts. Then we can transmit information quickly and do not need to inquire about the address of NS every time avoiding a potential bottleneck in system performance. In the same time, if the field “next” stayed time is too long it will affect the efficiency of message transmitting. So we set up a parameter T_0 with “timeout” in LS module. If a pointer’s stayed time is more than the maximum time T_0, then we set the field “next” to null.

Figure 3 Schematic diagram of the process of message:

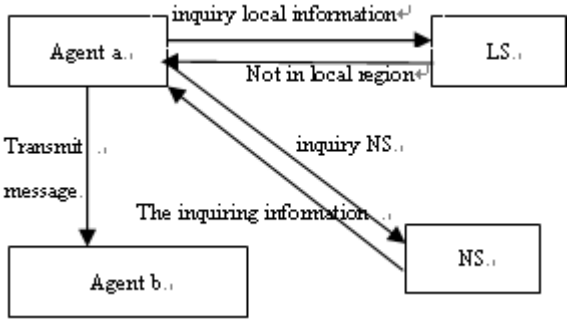

Figure 3 the process of the message forwarding

3.3 The de scription of solving the message chasing problem

In order to solve the problem of message chasing, firstly, we use three parameters T_0, N_0, V_0(N_0 is the times of message forwarding, V_0 is the maximal velocity of agent moving) to determine whether the message is the chasing message or not. Of course the relationship among them is the “or” relation. If the message is the chasing message, we must limit the mobile agent to stop and wait. Then in this paper we stored all messages which send to the goal agent in the mailbox module. The value of “mcount” field reduce by 1 when send a message. When the “mcount” of mailbox reduced the value of 0, the agent restores the previous state and can continue to move.

We assume T_n is the Communication delay time, Tm is the message forwarding processing time, the query of address cost $2T_n$, deal with the request of home address cost T_c, Tn far more than T_m.

4.2.1 Setting of parameters

We can calculate the message forwarding time according to the agent moved from node 1 to node n ,that is $T_1=nT_m+(n-1)T_n$; if the pointer stayed time exceed T_0, we can express $T_2=T_c+2T_n+2T_m+ T_n$. So we have come to an inequality ($T_1> T_2$), then to derive n> 4, that is, N_0= 4.

In order to chase the goal agent, the inequality ($V_1>V_2$) must be correct (V_1 is the moving speed, V_2 is the speed of message transmition), we know the value of V_2 is count/T, then there is $V_0=1/T_n$.

4.2.2 Agent strategy of limiting spending

Figure 4 is the description of the method “Agent strategy of speeding waiting” in document [1]. When the value of count is greater than N_0 or the value of V_1 is greater than V_0, we can the message is a chasing message. Then we set the field “tag” to true. Then control and stop the agent to waiting.

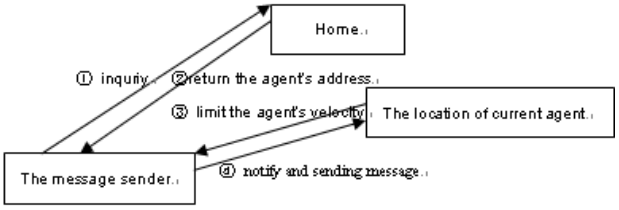

Figure 4 Agent strategy of speeding waiting

There exist two problems in the method of Figure 4. For example, it will be caused the message failure before the speeding limiting when the message chasing appeared; it also be too depend on the NS module when inquiry the NS. So we propose a novel method “Agent strategy of speeding limiting”, the proposed method can solve the two problems.

Figure 5 is the description of the “Agent strategy of speeding limiting” method. The mailbox can be used as a mobile agent’s communication agent, when the mobile agent move, it will notify the message sender to modify the communication message address as the mailbox node address, and send the message to the mailbox before the limiting speeding in order to reducing the loss of message.

Each step in Figure 5 describes as follows:

① First check the value of the variable “number”, if the value more than a setting value, then limit the agent’s speeding, set the next pointer to empty, so Agent is limited to its current location;

978-1-935068-01-3 © 2009 SciRes.

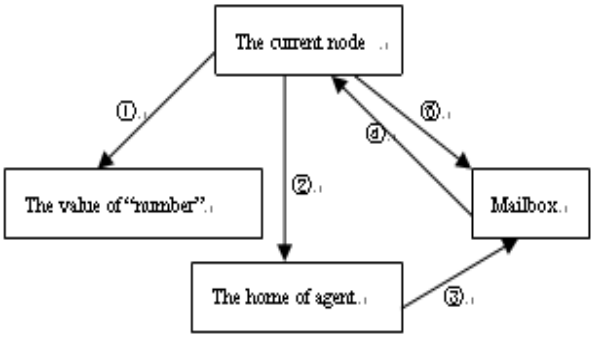

Figure5 Agent strategy of speeding limiting

② after the agent was stopped, it send the message to the home of its and also modify the address Home and LS to the current agent location address.

③ At the same time, notice the mailbox of agent A to send the all message in it, and to amend the destination name for the mailbox's current address;

④ Sending message to the current agent;

⑤ The value of "mcount" field reduce by 1 when send a message. When the "mcount" of mailbox reduced the value of 0, it indicates that the information sent to it is completed to receive.

4 Experimental results and Performance analysis

Table 1 is a simulation platform based on Aglets programming the data, Figure 6 is implemented in accordance with Table 1 and there is the density curve of comparing way to limit the message speeding with not limit. (Party, when N_0> 4 is the pursuit of news, it is necessary to limit). Suppose the value of T_n is 100ms, the V_0 is 0.01.

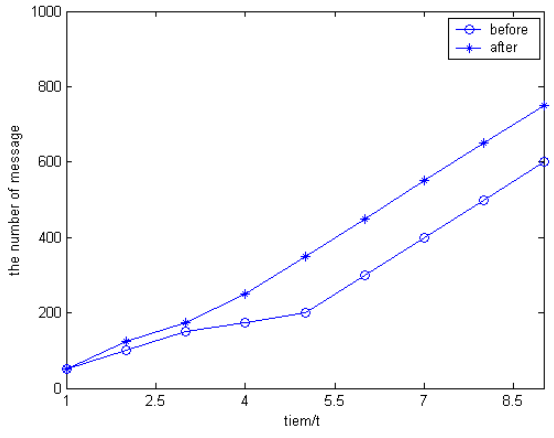

Figure 6 comparing with two methods

(1) This method can effectively avoid the situation of chasing, then to realize the reliability of message sending. As a result of NS stored all domains' address, so that the Agent will not move at random to receive message and also will be appear the phenomenon of chasing.

Table 1. System resulting data of standard expriment

Times of message forwarding	Agent moving speed	before	after
1	0.008	25	25
2	0.005	21.2	21.2
3	0.0011	15.3	0
4	0.003	4.9	4.0
5	0.004	1.8	0
6	0.005	1.5	0
7	0.003	0.8	0
8	0.004	0.5	0
9	0.005	0.02	0
10	0.008	0.03	0

(2) In this paper we use the pointer address scheme; it uses a centralized management to reduce the bottleneck. And the LS module can reduce the management of centralized structure.

(3) We take the synchronous communication on the question of message chasing, and we also use the mailbox module to manage the message which send to the agent.

5 Conclusions

In this paper we first judge the message whether is chasing message or not, then we take the agent strategy of limit strategy to solve the message chasing problem. In the communication model we take the sub-domain management, each region has a home module and LS module which has a field (next). The pointer can efficiently prevent the bottleneck caused by the frequently inquiries NS module. However, when restrictions on the mobile Agent location, we must inquiry the NS, so that there is also a potential bottleneck in the system. For future work, we need to adjust parameters we have presented above. In addition, we need to coordinate the mailbox forwarding message.

Finally, we plan to evaluate the performance reliability of the mobile agent and message sending.

Acknowledgements

This work is partially supported by “The research of foreign Chinese long-distance visualized teaching model” The National Society Science Foundation of P.R. China, Under Grant No. 07BYY03。The authors wish to thank all members of their research group for the supports of various aspects.

References（参考文献）

[1] ZHOU Jingyang, JIA Zhiyong and CHEN Daoxu, Designing Reliable Communication Protocols for Mobile Agents[M], Proceedings of the 23 rd International Conference on Distributed Computing Systems Workshops,2003.

[2] Amy L. Murphy and Gian Pietro Picco, Reliable Communication for Highly Mobile Agents[M], In Agent System and Architecture/Mobile Agents(ASA/MA) ’99, Oct 1999.

[3] FENG Xinyu,CAO Jiannong,et al, An Efficient Mailbox-Based Algorithm for Message Delivery in Mobile Agent Systems[J], Picco(Ed.), Mobile Agents,LNCS 2240, Spintger-Verlag, 2001,pp.135-151.

[4] Ranganathan M, Bednarek M, Montgomery D, A reliable message delivery protocol for mobile agents[J], Lecture Notes in Computer Science 2000;1882:206–20.

[5] Joachim Baumann, Fritz Hohl, Nikolaos Radouniklis, Markus Straßer and Kurt Rothermel, Communication Concepts for Mobile Agent Systems[M], In Mobile Agents, Proc. 1st Int. Workshop, MA`97. Springer, 1997.

[6] Raimundo J.Araújo Macêdo*, Flávio M.Assis Silva, Themobile groups approach for the coordination ofmobile agents[J], J. Parallel Distrib. Comput. 65 (2005) 275 – 288.

[7] S.Lazar,I.Weerakoon and D.Sidhu, A Scalable Location Tracking and Message Delivery Scheme for Mobile Agents[M], In Proc.7th IEEE Intl. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises ,1998.

[8] Andrew S.Tanenbaum and Maarten Van Steen, Distributed Systems Principles And Paradigms ,Prentice Hall.Inc[M].,2002.

[9] CAO Jiannong, FENG Xinyu, et al, Design of Adaptive and Reliable Mobile Agent Communication Protocols[M], In Proceedings of 22nd International Conference on Distributed Computing Systems(ICDCS 2002), July, 2002, Vienna, Austria.