Mechanisms to Locate Mobile Agents in Multi Agents Environment

Swati Aggarwal
Research Scholar
GKV Haridwar, India

Heman Pathak
Associate Professor
KGM Dehradun, India

ABSTRACT

Mobile Agent (MA) is autonomous software entity which moves from host to host in the open network like internet under its own control. It is once launched its creator or base host has no information about its location until it come back to the creator after completing its itinerary. In multi agent environment where multiple agents are launched, different MAs often need to locate each other in order to communicate, cooperate and to take decisions collectively. Mobile Agent (MA) location monitoring is a necessity in highly dynamic and large-scale mobile networks to control and communicate with agents after launching.

Effectiveness of any Location Management (LM) strategy mainly depends on the cost of searching and updating the database. This paper presents different mechanisms to locate MA in a multi agent environment with the objectives to reduce the search and update cost. Proposed approach divides the global network in regions. In each region there is a centralized component responsible to maintain the location of all MAs presently executing in its region. It also records the address of the region where MA is migrating. Various search and update schemes are then used to locate MA by its base host or other cooperating MAs.

Paper proposes alternative approaches for two different cases, first when Base Host of communicating MAs are same and second when Base Hosts for communicating MAs are different. Paper also proposes a mailbox based techniques to provide communication among the MAs.

KEYWORDS

Mobile Agent (MA), Mobile Agent System (MAS), Multi Agent System (MS), Location Management (LM), Base Host (BH).

1. INTRODUCTION

Mobile Agent (MA) is considered as the outlook of distributed computing. They are promising to offer a joined and scalable framework for applications in broadly distributed varied open networks, such as Electronic Commerce, Parallel Computing, Information Retrieval, etc. MA carries code and data with it. It moves from one host to other under its own control. For providing an execution environment, Mobile Agent System (MAS), installed at each host of the network.

Once MA is launched in the open network like Internet, its creator has no control over it and has no information about its location until it come back to the creator after completing its itinerary. For some MA applications, the ability of creator/Base Host (BH) to communicate with MA in real-time as it move from one network node to another is essential for retrieving any data or information that it has collected, to stop processing being done by the MA, to change its itinerary and for supporting co-ordination.

A Multi Agent System is a system composed of multiple interacting intelligent agents. It may contain multiple MAs launch by same host or by different hosts or combination of MAs and software agents. Multi Agent System is used to solve problems which are difficult or impossible for an individual agent or monolithic system to solve. For large scale multi agent environment multiple agents must be able to locate each other for supporting co-ordination and cooperation [1], [3], [4], [5], [6], [7]. In various situations, MAs at different hosts must cooperate with one another by sharing information and making decisions collectively [8]. Existing MAS must be enhanced to support multi agent system and must include the functionality to locate other MAs. This additional functionality can either be implemented as services offered by other MAs or as services more integrated into the MAS infrastructure itself. Examples of such services include facilitators [12], matchmakers, and mediators [11]. Most existing MASs currently do not provide a comprehensive, efficient or effective location management system. The support of these platforms for MA location ranges from limited to nonexistent. So, there is a need for a service with the ability to locate any type of mobile agent at anytime from anywhere.

Communication is an essential component of distributed systems and this is no exception for multi-agent systems [18]. Besides technical aspects of mobile agent technology, such as migrations [16] and control mechanisms, communication between mobile agents has been identified as an important issue in mobile agent technology [19]. This paper proposes the mechanism to locate MAs for both cases mentioned earlier as well as provide a communication mechanism between the MAs.

2. RELATED WORK

This section discusses the various existing approaches for MAs and for the Communication schemes among the MAs.

2.1 Location Management Schemes

There are different approaches for locating MA in the global network. This section discusses some of the existing location management schemes. In Database Logging protocol (DL) [14] location information of MAs is stored at a specific server called location database server, upon every migration. The location information is used to find the MAs. In path proxy protocol (PP) [14], MA leaves a forwarding proxy at the source node at every migration. MA is located by following chain of proxies. No location update procedure is used. The shadow protocol [14] is a combination of DL and PP protocol. A MA updates current location to an associated shadow according to TTL (time to live), which is a particular fix time interval. After the TTL, MA updates its current location to its shadow. In an optimal location update and searching algorithm for tracking MA, MA updates its location after it migrates through continuous d hosts since its last update. There is an optimal threshold d that makes the total cost of search and update minimized. If d assigned dynamically, it is called dynamic location update scheme [14]. Blackboard protocol [14]

maintains the blackboard at each node which is a shared information space for message exchange. When any MA wants to deliver a message, it puts the message in the blackboard. Afterword the receiving MAs move to the corresponding node where a message is stored and get the message. The SPC protocol [14] is a combination of DL protocol and PP protocol, applied to a multi-region MA computing environment. Location information is stored in a distributed way at a Region Agent Register (RAR) or a Site Agent Register (SAR). Agent is reached by following a part of the links that the agent has left on two registers. Number of location update operations is reduced in SPC protocol by applying an optimal location update and searching algorithm for tracking MA on SPC. The modified protocol is named as MBLM (Movement Based Location Management) [14]. This protocol is implemented on PMADE and found to have lesser cost of location update than SPC protocol. Scalable Hash-Based MA Location Mechanism [14] has special agents called information agents (IAgents). IAgents are responsible for maintaining the current location of a set of agents. Set of MAs associated with each IAgents is determined through the hash function. This association changes overtime as new IAgents are created or existing IAgents are merged depending on the system workload. A Home Blackboard (HB) protocol location management and message delivery protocol, concentrated more for confirmed message delivery. A HB protocol is a hybrid of a DL protocol and a Blackboard protocol [14] in order to complement each other and overcome.

2.2 Communication Mechanisms Among the Mobile Agents

In the literature different techniques are used for the communication of mobile agents. Mailbox techniques, Broadcast & black board were three most popular techniques used for communication. A mailbox [20], which is a message buffer used to store incoming messages destined to an agent, can be detached from its owner agent in the sense that the mailbox can reside at a location different from the current location of the owner agent. Every mobile agent in the system is allocated a mailbox. Incoming messages sent to the agent are inserted into the mailbox first. The communication between agents is divided into two steps: the *transmission* of a message from the sender to the receiver's mailbox and the *delivery* of the message from the mailbox to its owner agent. If an agent wants to send a message to another Agent, it simply sends the message to the receiver's mailbox. Later the receiver receives the message from its mailbox using either pull or push. In the push mode, messages stored in the mailbox will be delivered to the mobile agent, while with the pull mode, the agent fetches messages from its mailbox any time it decides to do so. [17]

Notice that, for a frequently migrating agent, its mailbox can migrate at a much lower frequency. There are three main aspects are for the MAs communications are: [21] The first approach is Frequency of Mailbox Migration. In this approach A Mobile Agent is always moving on the path but it does not carry its mailbox with itself. It leaves its mailbox at its Base Host. This technique is called No Migration (NM); or MA might carry its mailbox with its migration path. This pattern is known as Full Migration (FM); or MA might determine the location of its Mailbox dynamically upon each migration, this technique is known as Jump Migration (JM). The second approach is message Delivery from Mailbox-to-Agent. In this approach MA receives the messages from its mailbox using either a push or pull operation. In the Push (PS) mode, the mailbox keeps the address of its owner and forwards every incoming message to it. In the Pull (PL) mode, the mobile agent

keeps the address of its mailbox and retrieves messages from the mailbox whenever needed. The third approach is Synchronization of Migration-Delivery. In this approach Delivery of a message can be reliable according to the need of the user. If users demands high reliability, they can overcome message loss by (1) Synchronizing the Host's message forwarding and the Mailbox's migration (SHM), or by (2) Synchronizing the Mailbox's message forwarding and the Agent's migration (SMA), or (3) both, known as Full Synchronization (FS). NS denotes the extreme case of No Synchronization being performed.

3. HIERARCHICAL LOCATION MANAGEMENT SCHEME (HLMS)

As discussed earlier location management is one of the important issue related with MAs. It becomes essential requirements for mobile multi agent environment where communicating MAs must locate each other before communication starts. A location management protocol consists of location updates, searches and search updates. An update occurs when a mobile agent changes location. A search occurs when a mobile agent needs to be located. A search–update occurs after a successful search. The major issue with location management in mobile agent computing is the high cost associated with location update and search. The goal of an efficient location management strategy should be to provide low cost of location search and updates [23, 24]. The cost of a location update and search is characterized by the time taken for each operation, number of messages sent, size of messages, or the distance the messages need to travel. An efficient location management strategy should attempt to minimize the combined cost of the location Search and update. At one extreme, up-to-date information of the exact location of all mobile agents is maintained at each and every node in the network. This reduces the search time to locate the mobile agent. But each time the location of the mobile agent changes, a large number of associated location databases must be updated, which involves cost of location update. At the other extreme, no information is stored at any site of the network. To locate a mobile agent, a global search at all network sites must be initiated; however, when the mobile agent moves, no cost is associated with updating the location database. Between these two extremes, various approaches that balance the cost of search against the cost of updates are possible. [19]

This work is inspired by [19, 20, 2, and 10] and [21, 9]. All these approaches divide the network into regions and in charge of that region keep track of all mobile agents in each region. Search, update or combinations are then used to locate MA. Regions may be in one, two or multiple levels. Hierarchical location Management Schemes [2] assumes that internet is network of networks where networks are connected with each other via router. Each router maintains an Agent Table which contains all the MAs currently executing on the network and their *HostId*. When a MA moves from one host to other within the same network Agent Table is modified. When MA moves from one network to other, receiving router launch an Update Agent to inform the Base Host about the new network location of the MA. So in this approach Base Host always knows the current network of its MA. To communicate with MA, Base Host sends message to the MAs router, where router locate the MA by searching the Agent Table and deliver the message. In this way this approach combines both search and update schemes. This paper will extend this scheme for mobile multi agent where instead of base host, MAs launched by same host or different host locate each other before initiating the communication.

In the proposed Hierarchical Location Management Scheme (HLMS) for Mobile Multi Agent System, there are various components namely Router, Hosts, mobile agents and their mailboxes. Each host runs a Mobile Agent System (MAS) which is used for the execution of MAs. It also provides communication, mobility and security support for mobile agents. All the hosts are connected by the network and can communicate with one another by sending messages on the network. Here Paper uses the case of FM for the migration of Mailbox of MAs.

The different Components of Hierarchical Location Management System for Mobile Multi Agent are:

3.1 Router

It is a component which connects the different network with each other. All MAs are migrating from or entering into a local area network via router. Instead of just receiving and forwarding the MA, router in this mechanism plays an active role. The different component installed on the router is: [13, 15]

3.1.1 Location Management System (LMS)
It is a system which manages the location of the MAs.

3.1.2 Log Manager (LM)
It manages the arrival and departure time of the MAs.

Table 1: Component Description of the Hierarchical Multi Agent Model

Components	Description
Router/Network	Every network has a router and it is identified by *NetId*.
HostID	A host in the global network is identified by number known as *HostId*.
Mobile Agent	Every MA in the system is identified with unique *AgentId*. It carries Parent HostID and certificate of verification SC (String, Destination *HostId*) for communication and Another agent for communication.
Search Agent	In order to locate a MA, its base host launches a Search Agent (SA).
Update Agent	When a MA arrives at a network, an Update Agent (UA) is created to inform BH about the new network location of MA. It is also used to update the location to the Central repository.
Packet	Agents move from one part of the network to other via global network in form of packet. A packet consists of an Agent, sender *HostId*, destination *HostId* and type of agent MA, SA or UA.
Base Table	Every Network maintains a *BaseTable*, which is the list of *AgentId* and the current *NetId* of the MA.
Agent Table	Every network maintains an *Agent Table* which is the list of *AgentId* of MA and *HostId* of the hosting Node.
Log Table	For all MAs received at the router an arrival entry is logged in *Log Table* similarly before leaving a network departure entry is logged to the *Log Table*.

Agreement Table	It contains the *HostId* of the hosts with which agreement has been signed. It also contains the Date time and duration of agreement.

4. DIFFERENT METHODS OF LOCATION MANAGEMENTS AND COMMUNICATION

4.1 Case 1- For the Same Base Host
This section proposes a mechanism to locate different MAs launched by same Base host. Every host maintains a Base Table which stores network Ids of all the active MAs launched by it. Base Host always knows the current network location of MAs.[18] So if a MA wants to communicate with other MA launched by same base host, it initiates four processes: -

1. **GetNetId:** This process launches a MA to visit the base host to collect the NetworkId of the target MA.

2. **ReturmNetId:** This process returns the Network id of the target MA.

3. **GetHostId:** This process initiates a search procedure to visit the NetworkId of target MA and search the local Agent Table to get the HostID of target MA.

4. **ReturmHostId:** This process returns the Network id of the target MA.

5. **SendMessage:** This process is initiated when mailbox address of the target agent is known, it drops the message to the mailbox of the target MA.

6. **GetMessage:** This process is initiated by the target MA. Here target MA wants to get the message from its mailbox. To get the message MA use push and pull approach as shown in figure1. In the Push (PS) mode, the mailbox keeps the address of its owner and forwards every incoming message to it. In the Pull (PL) mode, the mobile agent keeps the address of its mailbox and retrieves messages from the mailbox whenever needed.
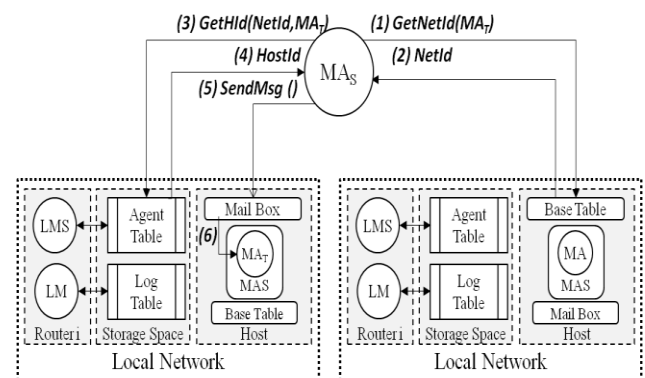

Fig 1 : - MA communication for the same Base Host

4.2 Case 2: - For Different Base Host
4.2.1 Two way communication
Sometimes there is a need to communicate among different MAs launched by different hosts. MAs launched by two different hosts can communicate with each other only if both the hosts trust each other and have signed an agreement to allow their respective MAs to locate and communicate with each other. For signing an agreement every host contains and Agreement Table which contains the HostID of the signed Host Through agreement each MA carries with it the Base Host of

itself as well as the HostId of the Host with which agreement has been signed.

According to [2] every host maintains a Base Table which stores the *AgentId* and its current *NetId*. Hence to communicate with any MA, source MA requires the current *NetId* of target MA. Since it already knows the Base Host of the target MA it will send a request to the Base Host of target MA to get its current NetId. For the security purpose it should not be allowed for any MA to locate and communicate with any other MA. When MA send a request for the *NetId* to the target Host then target host will only allow access of *NetId* to the trusted MAs.

So if a MA wants to communicate with other MA launched by the different base host, it initiates six processes: -

1. **GetNetId:** This process launches a MA to visit the base host of target MA to collect the NetId of the target MA.

2. **ReturnNetId:** This process returns the Network id of the target MA.

3. **GetHostId:** This process initiates a search procedure to visit the NetworkId of target MA and search the local Agent Table to get the HostID of target MA.

4. **ReturmHostId:** This process returns the Network id of the target MA.

5. **SendMessage:** This process is initiated when mailbox address of the target agent is known, it drops the message to the mailbox of the target MA.

6. **GetMessage:** This process is initiated by the target MA. Here target MA wants to get the message from its mailbox. To get the message MA use push and pull approach as shown in figure1. In the Push (PS) mode, the mailbox keeps the address of its owner and forwards every incoming message to it. In the Pull (PL) mode, the mobile agent keeps the address of its mailbox and retrieves messages from the mailbox whenever needed.
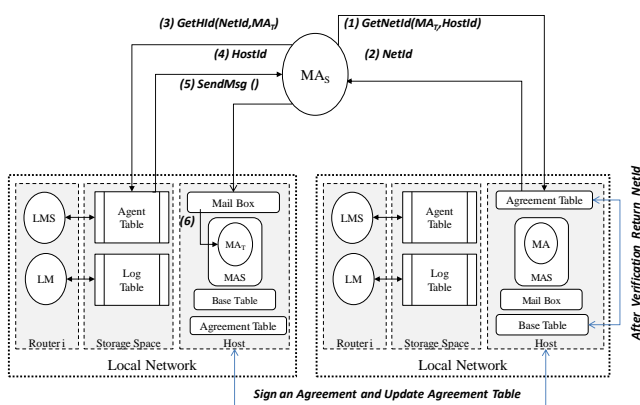

Fig 2: - MA communication for the different Base Host

But this method has a limitation of high cost associated with the tracking of location of MA. It is a kind of two way communication between the different MAs which increase the cost of searching and update.

5. CONCLUSION

Communication determines how much information is exchanged between agents. The amount of information available to an individual agent will in turn influence the decision-making process. In open MAS, agents are mobile and may leave or enter the system. This dynamicity results in two closely related agent communication problems, namely efficient message passing and service agent discovery. For the communication, track the location of MAs are very important. This work is based on this approach. Here author gave two different ways of communication for MAs.

Global network is the network of networks. The different networks are connected with each other via Router. All MAs are migrating from or entering into a local area network via router. Host can be the part of any network in the global network. If Base host of different agent are same so there is no need to find the *NetId* of the MA because Base Host already knows it but if Base Host are different, first Both the host sign an agreement between each other then the agent can communicate. But in this case searching cost is very.

6. FUTURE WORK

This work is based on the location of MA's and their communication in a multi agent environment. Here paper has explained the different cases of Mobile agent communication. Paper work will simulate in the next paper using CPN tool. The work o this is going on and simulation gives a better result in terms of cost. Paper will also propose some solutions and problems of these approaches.

7. REFERENCES

[1] J. Chen, "A Hierarchical Fault-Tolerance Framework for Mobile Intelligent Agent Systems," Master of Science thesis, The University of British Columbia, Faculty Of Graduate Studies, Department of Computer Science, April 2002.

[2] Heman Pathak, "Comparative Performance of Hierarchical Location Management Mechanisms for Mobile Agents" in Journal of Information Systems and Communication (JISC), ISSN: 0976-8742, E-ISSN: 0976-8750, Volume 3, Issue 1, pp: 278-281, 2012.

[3] E. Gendelman, L. F. Bic, and M. B. Dilllencourt, "An Application transparent, platform independent approach to rollback-recovery for mobile agent systems" 20th IEEE International Conference on Distributed Computing Systems, Taipei, Taiwan, pp. 564-71, April 2000.

[4] D. Johansen, K. Marzullo, F. B. Schneider, K. Jacobsen, and D. Zagorodnov, "NAP: Practical fault-tolerance for itinerant computations", In Proceedings of the 19th IEEE International Conference on Distributed Computing Systems (ICDCS'99), Austin, Texas, USA, pp. 180-189, June 1999.

[5] D. Kotz, R. Gary, "Agent Tcl: Targeting the Need of Mobile Computer", IEEE Internet Computing, pp. 58-67 July/August 1997.

[6] R. Michael, T. Y. Wong, "A Progressive Fault Tolerant Mechanism in Mobile Agent Systems," in Proc. of the 7th world Multi-conference on Systematics, Cybernetics and Informatics, Vol. IX, Orlando, Florida, July 2003, P.P. 299-306

[7] S. Mishra, Y. Huang, "Fault Tolerance in Agent-Based Computing Systems," Proceedings of the 13th ISCA International Conference on Parallel & Distributed Computing, Las Vegas, N V. August 2000.

[8] R. B. Patel, K. Garg, " Fault-Tolerant Mobile Agents Computing On Open Networks", www.caip.rutgers.edu/~parashar/AAW-HiPC2003/patel-aaw-hipc-03.pdf

[9] "Tracker: A Universal Location Management System for Mobile Agents" George Samaras, Constantinos Spyrou, Evaggelia Pitoura+, Marios Dikaiakos Department of Computer Science, University of Cyprus CY-1678 Nicosia, Cyprus, cssamara@cs.ucy.ac.cy + Department of Computer Science, University of Ioannina GR 45110, Ioannina, Greece, pitoura@cs.uoi.gr

[10] G. Serugendo and A. Romanovsky, "Designing Fault-Tolerant Mobile System", Springer, International Workshop on Scientific Engineering for Distributed Java Applications, 2003, pp 185-201, ISBN: 978-3-540-00679-4

[11] S. Pleisch and A. Schiper, "FATOMAS-A Fault-Tolerant Mobile Agent System Based on the Agent-Dependent Approach", IEEE, Proceedings of the 2001 International Conference on Dependable Systems and Networks, 2001, pp 215-224, ISBN:0-7695-1101-5

[12] In 2002 Stefan Pleisch, "Fault-tolerant and transactional Mobile agent execution."

[13] Heman Pathak, "Search and Update Based Solutions to Locate Mobile Agents in the Global Network" in International Journal of Mobile And AdHoc Network (IJMAN), ISSN(Online): 2231:6825 ISSN(Print): 2249–202X, Vol. 4, Issue 1, Pp: 75-79, Jan-Apr 2014.

[14] Charu Virmani, "A Comparison of Communication Protocols" International Journal of Advancements in Technology http://ijict.org/ ISSN 0976-4860

[15] R. B. Patel, Nikos Mastorakis, GREECE K. Garg, "Mobile Agent Location Management in Global Networks , Department of Computer Science & Engineering M. M. Engineering College, Mullana- 133203, Haryana, India,

Dept of Computer Science, Military Inst. of University Education / Hellenic Naval Academy Terma Hatzikyriakou 18539, Piraeus, Department of Electronics & Computer Engg, IIT Roorkee, Roorkee- 247667, Uttaranchal, India

[16] Heman Pathak, "Colored Petri Net based Modeling of Hybrid Location Management Mechanism for Mobile Agents" in Journal of Information Systems and Communication (JISC), ISSN: 0976-8742, E-ISSN: 0976-8750, Volume 3, Issue 1, pp: 291-294, 2012.

[17] Xinyu Feng1, Jiannong Cao2, Jian Lü1, and Henry Chan2, "An Efficient Mailbox-Based Algorithm for Message Delivery in Mobile Agent Systems" G.P. Picco (Ed.): MA 2001, LNCS 2240, pp. 135-151, 2001. © Springer-Verlag Berlin Heidelberg 2001

[18] Swati Aggrawal, Heman Pathak, "Analysis of Location Management Mechanisms forMobile Multi Agents" in International Journal of Information and Computation Technology(IJICT), ISSN :0974-2239, Volume 3, Number 9pp: 917-926, 2013.

[19] Rama Sushil, Rama Bhargava, Kumkum Garg" Location update schemes for mobile agents" 1Asst. Prof. SGRRITS D. Dun, R/S Dept. Of Maths, I. I. T. Roorkee, India, ramasushil@yahoo.co.in,2Prof. Of Mathematics, I. I. T. Roorkee, India, bhargava_iitr@rediffmail.com,3Prof. Of Computing, I. I. T. Roorkee, India, Sr. Member IEEE, kgargfec@iiternet.in

[20] R. B. Patel, Nikos Mastorakis, K. Garg " Mobile Agent Location Management in Global Networks" Department of Computer Science & Engineering,M. M. Engineering College,Mullana-133203 Haryana, India Dept of Computer Science,Military Inst. Of University Education / Hellenic Naval Academy Terma Hatzikyriakou 18539,Piraeus, GREECE, Department of Electronics & Computer Engg, IIT Roorkee, Roorkee-247667,Uttaranchal, India

[21] Jiannong cao, liang zhang, xinyu feng+ and sajal k. Das++ "path pruning in mailbox-based Mobile Agent Communications", JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 20, 405-424 (2004)