International Journal of Information and Computation Technology.
ISSN 0974-2239 Volume 3, Number 9 (2013), pp. 917-926

http://www. irphouse.com /ijict.htm

Analysis of Location Management Mechanisms for Mobile Multi Agents

Swati Aggarwal[#] and Heman Pathak[*]

*[#]Department of Computer Application-*Department of Computer Science,*
[#]IMS Engineering College Ghaziabad.
[*]KGM, Dehradun.

Abstract

Mobile Agent is a process that can transport its state from one environment to another, with its data intact, and still being able to perform appropriately in the new environment. For some applications, instead of one Mobile Agent, group of mobile agents are launched. These MAs are executed independently across the open network like internet but often required to interact with each other to know the status of other MAs and to share data/partial results. Sometimes there is a need to track the location of Mobile agents so location management and location tracking of a mobile agent is an important issue with the mobile agent technology. Similarly in multi agent environment where multiple agents are working simultaneously, tracing the location of mobile agents is very difficult. So, Location Management (LM) of mobile agents has become an assorted and lane field for research. This paper presents various approaches to locate mobile agents in global network after launching. Paper also attempts to analyse these approaches by computing the cost of locating MAs in terms of the time required for basic operations search and update. Based on the result favourable approaches have been suggested for different cases such as Frequent Operation, Rare Operation and Base Host Heavily loaded.

Keywords: Location management strategy, Mobile agent, mobile agent system, Multi agent System, Mobile agent host, Cost of location update.

1. Introduction

Due to the demand for better technology and extensive research in the field of distributed computing, new techniques, languages and paradigms have emerged. Among them are traditional client/server architecture, Remote Method Invocation (RMI) and Remote Procedure Call (RPC). But, for certain applications, existing technologies limit the flexibility with which a user uses the resources/services offered across the network. The latest and most promising paradigm in the field of distributed computing is Mobile Agent (MA) technology. For some applications, instead of one Mobile Agent, group of mobile agents are launched. These MAs are executed independently across the open network like internet but often required to interact with each other to know the status of other MAs and to share data/partial results. For large scale multi agent environment multiple agents must be able to locate each other for supporting co-ordination and cooperation [1], [2], [3], [4], [5], [6]. This paper proposes different approaches to locate MAs in a Multi Agent environment. Basic operations involve to locate MAs are Search and Update. Proposed approaches are efficiently uses these operations to get better results for different case scenario. Some suggestions to improve the performance in case of frequent or rare cases and for heavily loaded hosts have also been suggested.

Mobile agents move from one node to another. Often, mobility is ad-hoc; [7] that is, the MAs move autonomously and asynchronously, without following a predefined route. Sometimes the creator of an agent needs to keep track of the agent's current location, in order to contact the agent and get access to its data and resources. Furthermore, other agents, called client agents, may need to contact the agent. In this case information retrieval is almost impossible, even if the agent follows the predefined route. The location problem becomes even more difficult when agents need to locate and cooperate with agents in other execution environments. Thus there is a need for a service with the ability to locate any type of mobile agent at anytime from anywhere. Such a mechanism (i.e., TRACKER) must be generic, flexible, independent of the agent platform and able to dynamically accommodate location algorithms to support various moving and invocation patterns. Some of the research in this area includes the Database logging protocol (DL), path proxy protocol (PP), shadow protocol, Blackboard protocol, Hash-Based MA Location Mechanism [7] etc.

2. Assumptions and Requirements

MA location is based on some assumptions for the distributed environment as shown in Figure-1. Global network is the network of networks. These networks are connected with each other via Router. All MAs are migrating from or entering into a local area network via router. Mobile Agent System is installed at each router, but it is responsible only to receive and forward them not to execute them. Router is assumed to be fault free. In each local area network, there are various agent hosts to create and execute MAs. Model also assumes the availability of a fault free local shared storage space within each local area network to store some shared information. Each MA

created as well as all hosts in the global network are assumed to be identified by unique MA_id and host_id. MA carries with it the address of its base host. Some other assumptions are -

1. All hosts are equally distributed over the Internet.
2. Time require to transmit a package from one host to other depends only on the type of package and it is independent of location of hosts i.e. equal time require to transmit a package from any host to any other host.
3. Traffic delay has been ignored. All hosts or networks and links are fault free and trust worthy i.e. all packages reach to its destination without being altered.

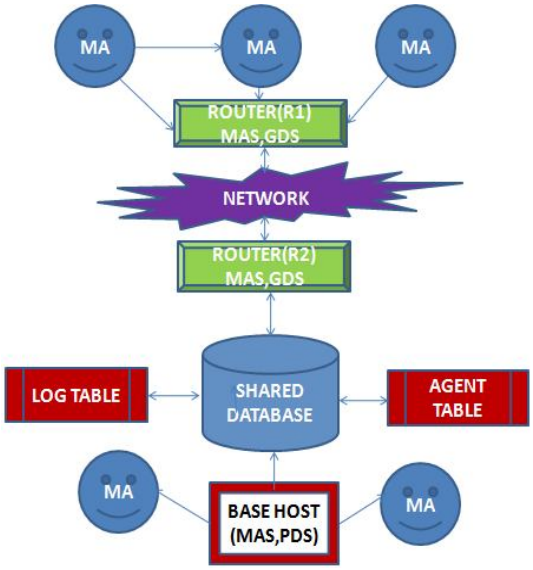

Fig. 1: System Model for location management approaches.

3. Location Management Approaches

Here we propose an easy to implement mechanism for the controlling of MA's. This mechanism contains three different mechanisms:

Frequent operation: - Sometimes it is necessary to track the location of an agent hence this operation becomes a frequent operation. We can subdivide it according to the different hosts who are responsible for the launching of the agent. We have two different options available.

For the same user: - In a multi agent environment different agent wants to communicate with each other so for the communication location tracking is a great issue. How we can trace the location of an agent. When a host launches multiple mobile agents and these agents wants to communicate with each other at that time the ***Always update*** method will be used by the agent to

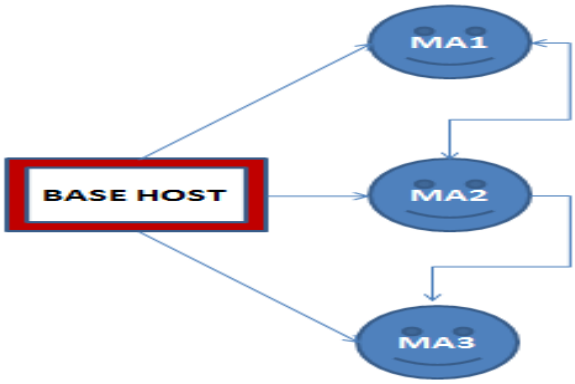

Fig. 2: MA communication for the same user.

track the location of an agent. In this process suppose a Base Host Launch multiple mobile agents and the mobile agents of the same network (**for example MA1 and MA2 as shown in fig**) want to communicate with each other then they can easily track the location of each other because BH knows the **current network address of each node**. So BH will not perform a search operation to locate MA. For the location tracking MA send a request to the BH for finding the location of another MA then BH will search the AgentTable to find the MA's host location. Here Update is a very common or frequent operation. Here update operation is enough to track the location. [9] This method is very easy and fast for tracking the location of MA but it is very inefficient in terms of cost because of every update operation of all MA in the network. [8]

For the different user:- Similarly in a multi agent environment different agent wants to communicate with each other so for the communication location tracking is a major issue. When a host launches multiple mobile agents and these agents wants to communicate with each other at that time the **Always update** will be used by the agent to track the location of an agent. It describes how the MA updates its Base Host after every transmission. In this process let us suppose a different Host launches multiple mobile agents at the same time and the mobile agents launch by different host of different network (**for example MA1 & MA4 or MA3 & MA5as shown in fig**) wants to communicate with each other, for this through Router2 this request will be handle. Router 2 sends a message to Router1 for the communication request by MA. Then Router1 sends this request to BH and BH provides the actual the host location of the MA and again BH will find the Agent Table to know the MA id and Host location. Because AgentId of MA is unique so it is easier to track the location. But again in this approach the cost is very high.

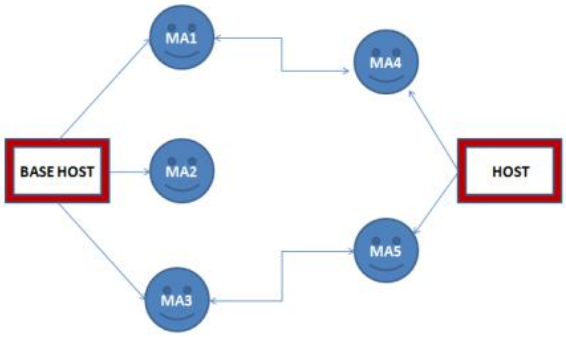

Fig. 3: MA communication for the different user.

Rare operation: - It describes when and how the BH will search the location of MA.

For the same user: - In this process suppose a Base Host Launch multiple mobile agents and the mobile agents of the same network (**for example MA1 and MA2 as shown in fig**) want to communicate with each other then they cannot easily track the location of each other because BH does not knows the current network address of each node. Because every time MA leaves its trace on every node that it has visited and tells about its next location. For the location tracking MA send a request to the BH for finding the location of another MA in the same network then BH will launch a search agent to find the MA's host location. [9] This method is very cost effective because there is no need to update every time but only when required BH can search a MA.

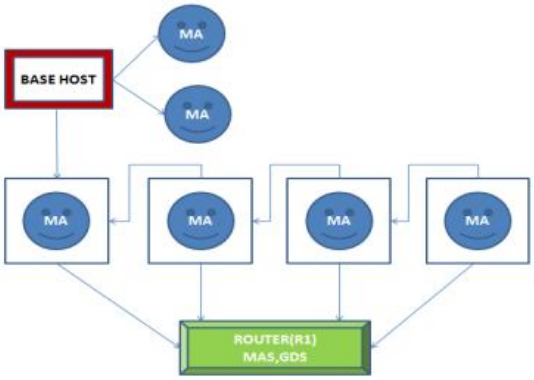

Fig. 4: MA search operation for the same user.

For different user:-In this process let us suppose a different Host Launch multiple mobile agents at the same time and the mobile agents launch by different host of different network (**for example MA1 & MA4 or MA3 & MA5as shown in fig**) wants to communicate with each other, for this through Router2 this request will be handle. Router2 sends a message to Router1 for the communication request by MA. Then Router1 sends this request to BH and BH launch a search agent to find the actual host location of the MA.

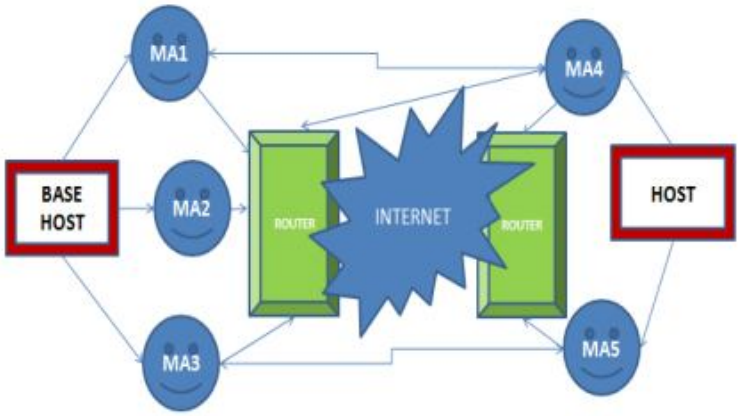

Fig. 5: MA search operation for the different user.

Base host is heavily loaded:-

In this approach we can use a mail Box technique to store the actual location of the MA. Some times after every Update operation and Search operation BH becomes heavily loaded so rather than Update and Search operation we can use a Mail Box technique. Every Router has their mail Box. When MA update it's location then every time an entry has done in the mail box. So it can decrease the load of BH.

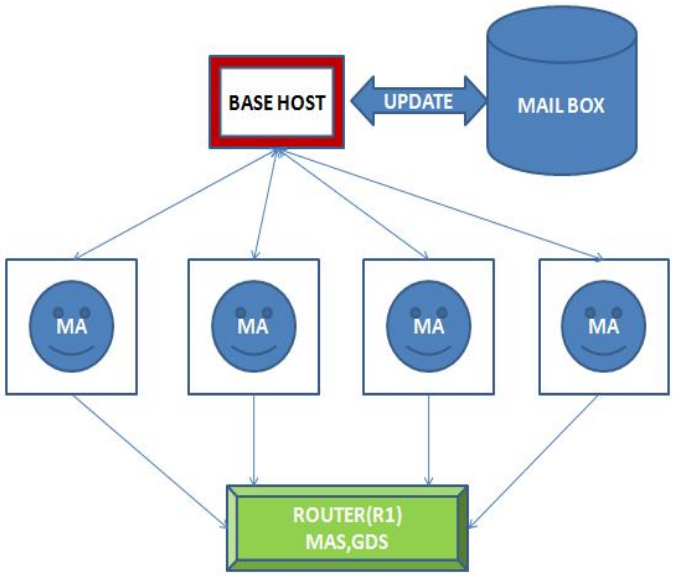

Fig. 6: Mail Box Approach.

4. Proposed Work

All MAs are migrating from or entering into a local area network via router. Here Router is a working entity. A HOST can launch a Single or Multiple Mobile agents at

the same time. We are dealing with the Multi Agent System so different agents can interact with each other but for interaction they need to know their location some time. Here we use the method of Always Update and Always Search. [3] Network architecture shows the communication between different mobile agents of the different network. Update and search procedure is used to find the location of the MA. Every time a MA is received at the router, GDS launches an UA to inform the BH of MA about its new Network address. LogTable is used to log an entry for arrival and departure of MA to the network. AgentTable stores the AgentId of MA and HostId of the hosting Node. When MA migrates from one host to other with-in the network, only Agent Table entry is updated. Personal Daemon Server (PDS) employed as a proxy server and it is installed at each host and watches the activities at host. When base host wants to locate MA, it need not to perform search as it always knows the current network location for MA, but to get the current host it has to search the Agent Table. Log table is implemented on the local shared storage space to log arrival and departure of every MA within the network.

5. Analysis

This mechanism is better than the previous approaches as it reduces number of search and updates. Since it takes advantage of existing infrastructure, it performs better than other region based location management mechanisms [10, 11]. A Mobile Agent (MA) plays the role of replacement of a user in the network. Suppose a MA and Destination Host are at the two ends of a communication link, and the another MA is placed somewhere in between these two ends. Naturally, the Second MA should be placed close to the first MA for communication. But it can also be at a far distance. Therefore, whenever the MA moves, it moves along. This approach is unsuitable when the MA is highly mobile, as too much movement of the MA incurs overhead. Therefore, if a mobile agent goes to at a very far distance in the network, then it's very difficult to track it in the network. To find the location of a Mobile agent in the network the researcher has discussed the two approaches named Always Search and Always update. Here we combined both the approaches in a single approach for different network and tried to make it a better approach.

6. Conclusion

Here the presented work shows a general overview of requirements for the need of location management in a multi agent environment. Support is needed for mobile agent location management in terms of efficiency and cost. This paper is based only on the different mechanism of finding the location of MA in the multi agent environment. These mechanisms can be much faster than the always search mechanism. The lack of efficient location update schemes of mobile agent system made the researcher in [12] to propose a basic dU −SSM scheme for tracking mobile agent. That is a mobile agent Updates it location once every d movements and then be searched by Sequential Searching Method. In paper [12], they analyze the agent’s mobility features and

formulate its cost functions based on a single agent's itinerary. They solve the optimal d functions by minimizing the total cost of location update and searching. By applying the proposed algorithm they obtain very good numerical results. We will also using the same approach of finding the location. Still we are working on the simulation and analysis part of it. Hence improve the location management of MA by using the existing infrastructure without increasing the overhead.

Reference

[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] 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.

[3] 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.

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

[5] 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

[6] 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.

[7] *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

[8] Heman Pathak," Comparative Performance Of Hierarchical Location Management Mechanisms For Mobile Agents" Department of Comp. Sc. KGM Dehradun, India.

[9] Patel R.B., Garg K. (2003) Autonomic Applications Workshop in conjunction with the International Conference on High Performance Computing, 17-20, 296-308.

[10] Pathak H., Garg K., Nipur (2004) National Conference on Trends of Computational Techniques in Engineering, 119-123

[11] H. Pathak, K. Garg, Nipur, "*Fault Tolerance Approaches for Mobile Agent Systems: A Parameter Based Comparative Study*", In proceedings 2004 National Conference on Trends of Computational Techniques in Engineering (TCTE '2004), P.P. 119-123, October 2004.

[12] Tie-Yan Li, & Kwok-Yan Lam, "An optimal location update and searching algorithm for tracking mobile agent", Information Security Group, Ubiquitous Computing Program, Laboratories for Information Technology, Singapore 119613 & litieyan@lit.org.sg, Centre for Systems Security, School of Computing, National University of Singapore, Singapore 119260, lamky@comp.nus.edu.sg