



A Comparison of Migration Strategy of Mobile Agent System

¹Sarita Rani, ²Er. Sahil Batra

¹Geeta Institute of Management and Technology Kanipla Kurukshetra, India

²Assistant, Prof., Deptt of Computer Engg. Geeta Institute of Management and Technology Kanipla Kurukshetra, India

Abstract— Mobile Agents are autonomous and proactive software entities which act on behalf on an owner and have the ability to migrate through a heterogeneous network of computer .A mobile agent can also decide when to migrate and which node to access. In the past few years, mobile agent (MA) paradigm has received a great deal of attention. While MAs have generated considerable excitement in the research community, they have not translated into a significant number of real-world applications. One of the main reasons for this is the lack of work that quantitatively evaluates the effectiveness of mobile agents versus traditional approaches. Of course, while the technology is indeed impressive, it also has raised some concerns about migration cost of mobile agent. This paper provides a comparison of migration strategy of mobile agent.

Keywords—Mobile Agent, Code Migration, Itinerary, Migration cost, push, pull

I. INTRODUCTION

A mobile agent is a program, which represents a user in a computer network, and is capable of migrating autonomously from node to node, to perform some computation on behalf of the user. Mobile agents are defined as objects that have behavior, state, and location [2]. Its tasks are determined by the agent application, and can range from on line shopping to real-time device control to distributed scientific computing. Applications can inject mobile agents into a network, allowing them to roam the network either on a predetermined path, or one that the agents themselves determine based on dynamically gathered information

II. MOBILE AGENT MIGRATION STRATEGY

According to LIU Da-you[8] , Yang Bo Mobile Agents are autonomous and proactive software entities which act on behalf on an owner and have the ability to migrate through a heterogeneous network of computer .A mobile agent[4] can also decide when to migrate an which node to access. After migration to the desired host the mobile agent resume the execution of previously broken off or awaiting task. On completing the task, the mobile agent returns the result to the client. Therefore the client need not be constantly connected to the server. This will not only save lots of unnecessary transmission load. But also helps in the application of mobile calculation. The mobile agent can single execute all task assigned by the user. It can meet and interact with other agent when necessary while still executing its task. According to LIU Da-You, Yang Bo 2003 there are 3 migration strategies.

(A)Ideal Migration Strategy: it is also called static migration. In this approach, a explicit itinerary graph is created based on its initial itinerary, after creating itinerary graph load information of itinerary graph is fetched. The drawback of this approach is that the change of load will outdate the edge and it requires a centralize server.

(B)One step Migration: This approach finds an optimal solution step by step during migration. It considers only the load on adjacent vertex. The drawback of this approach is that it does not provide the global optimal solution.

(C)Learning Migration Strategy: This approach tries to find out a globally optimal solution, in this approach during each time, agents records the software and hardware load information of a network. Next time, agent can use those travel experiences accumulated during previous migration. Each time, the old experiences will be updated by new information collected by agent. The drawback of this approach is that it does not always provide globally optimal solution.[7]

III. ITINERARY PLANNING FOR MOBILE AGENT

According to Christian Erfurth, Wilhelm Rossak[7] A Mobile agent visit more than one host in a network which is called agencies to fulfill its task. Thus it needs a travel plan when its starts its journey, a usually fixed plan is provided by its owner. The plan should be dynamically changes when mobile agent moves from one host to another.

(i)Map Module

It is responsible for collecting data to create a local information base which is called map. Every agency manages its local map. Because of the given quantity it is not possible to store complete information of the network of all agencies. Hence the map is split into a local area map and a map of remote areas.

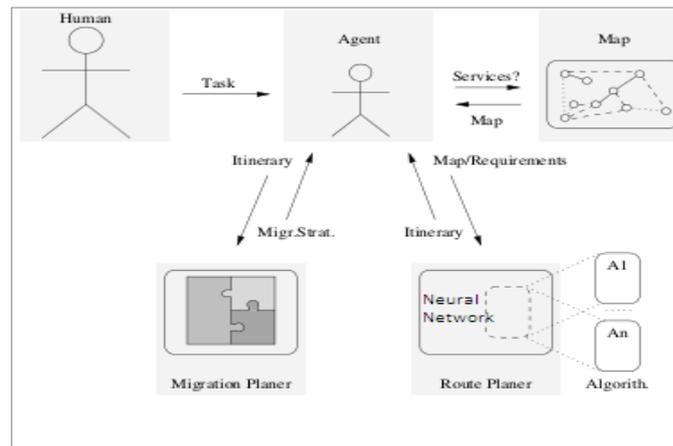


Fig1: Architecture of Itinerary Planning

(ii)Router Planner Module

The Route Planner Module is able to calculate a shortest path through a network. This module uses the map data, especially the data on connection topologies and qualities.

(iii)Migration Planar Module

It is designed to reallocate the itinerary to achieve an efficient journey. To resume execution of a migrated agent on a remote agency, the agent code with all referenced code parts is needed as well as the agent’s execution states and its current data. [8]

IV. CLASSIFICATION OF ITINERARY

According to Susmit Bagchi Itinerary is the set of sites that MA has to visit, it can be classified as:

1. Static
2. Dynamic

Order:Based on the visiting pattern order can be:

1. Static
2. Dynamic

Types of Itinerary;

1. Static Itinerary Static Order (SISO)
2. Static Itinerary Dynamic Order (SIDO)
3. Dynamic Itinerary Dynamic Order (DIDO)

If we know the different server address then it is best to use the static itinerary agent. If we do not know the server address then it is better to use the dynamic itinerary agent.

1. Static Itinerary Static Order (SISO)

In static itinerary the list of the remote host address is given by the owner at the time of dispatching the agent, the mobile agent should visit only the listed remote host and return to its home.

In SISO, the agent should visit only the given remote host in the given order.

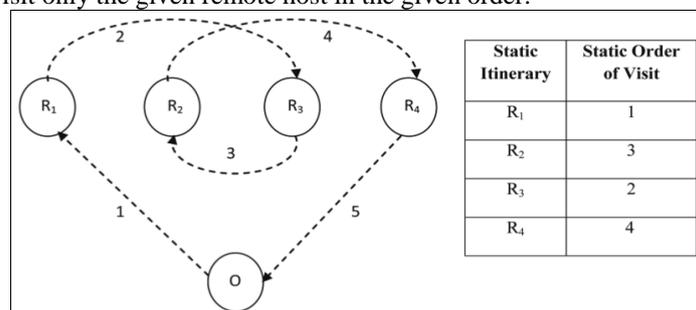


Fig 2: Static Itinerary Static Order

2. Static Itinerary Dynamic Order (SIDO)

In SIDO , the agent should visit the given list of remote host in the dynamic order .The order of visiting the remote host is decided based on the current conditions (shortest path or network traffic based routing) of the host, where the agent is currently residing.

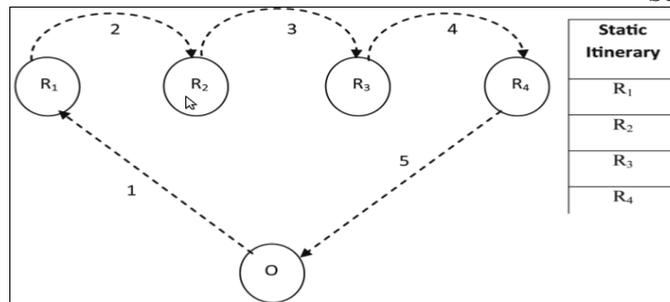


Fig. 3 Static Itinerary Dynamic Order

3. Dynamic Itinerary dynamic Order (DIDO)

In DIDO mobile agent, the client only knows the very first remote server for sending the mobile agent whether the required content is there or not. The remaining remote server will be visited with the help of the server where the agent is currently residing that is based on the requirement of the agent.

DIDO mobile agent will roam with two different things.

1. Query Based
2. Non Query Based

1. Query Based Approach

In query based approach the client will send the mobile agent with the query for the particular content to the first remote server. The first remote server will be selected with the name.

If the content is there than agent will collect the information otherwise it will not get any information and ask the help of the server to transfer it to the next server.

2. Non Query Based Approach

In Non query based approach it will ask the server to forward it to the next server which is having the information relevant to the query (in case of query based) otherwise it will ask the server to forward to the nearby servers .

After visiting the particular number of servers (decided by the client), agent will return back to the home (client) with the information.[6]

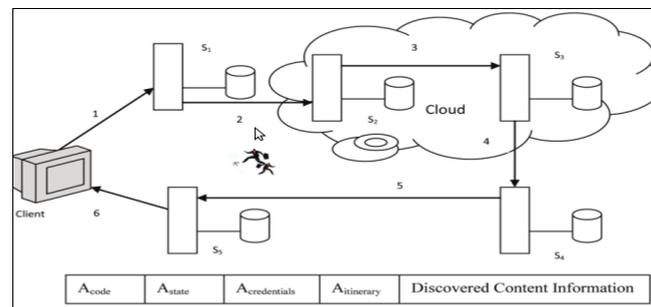


Fig.4: Non Query Based Approach

According to Peter Braun[5] some migration approach can be classified into the push and pull category.

a) Push-all-to-next

It transfers entire code to the next host, while sending the state of the mobile agent. This approach is virtually dependent on the size of the code and does not impose a continuous load.

b) Push-all-to-all

It transfers entire code of agent to all the host of a network. In this scheme mobile agent should know all the destination before transmitting the code. When the MA arrives on the destination platform, MA's execution can start immediately without any further code transmission.[4]

Table 1. Push –all-to-next without cache vs. using caching mechanism

Size of MA Code (byte)	Migration Cost without cache(ms)	Migration cost with cache(ms)	Performance ratio (%)
7.5K	65	55	118
64K	140	50	280
160K	275	47	585

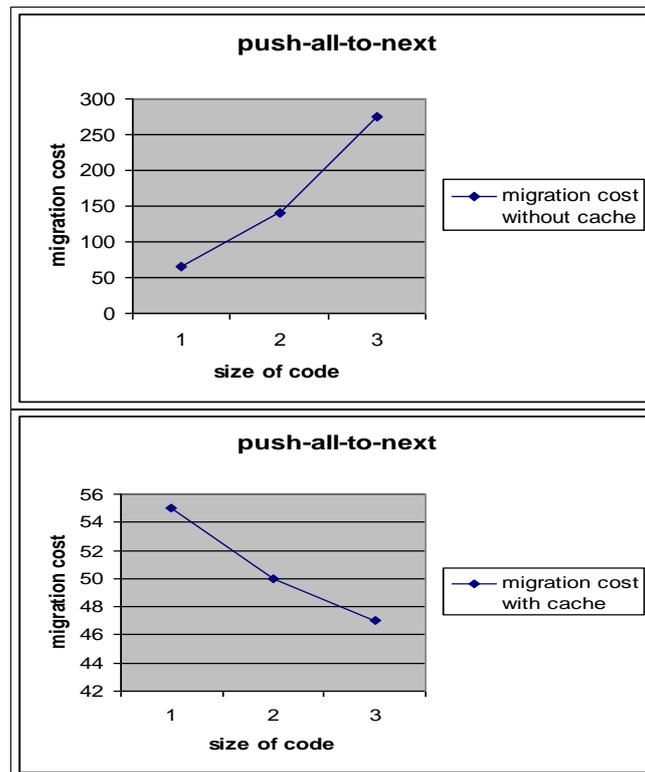


Fig.5: push all to next

Table 2 Push-all-to-all without cache vs. using cache mechanism

Size of MA code (byte)	Migration cost without cache(ms)	Migration cost with cache(ms)	Performance ratio(%)
7.5K	170	55	309
64K	430	70	614
160K	840	104	807

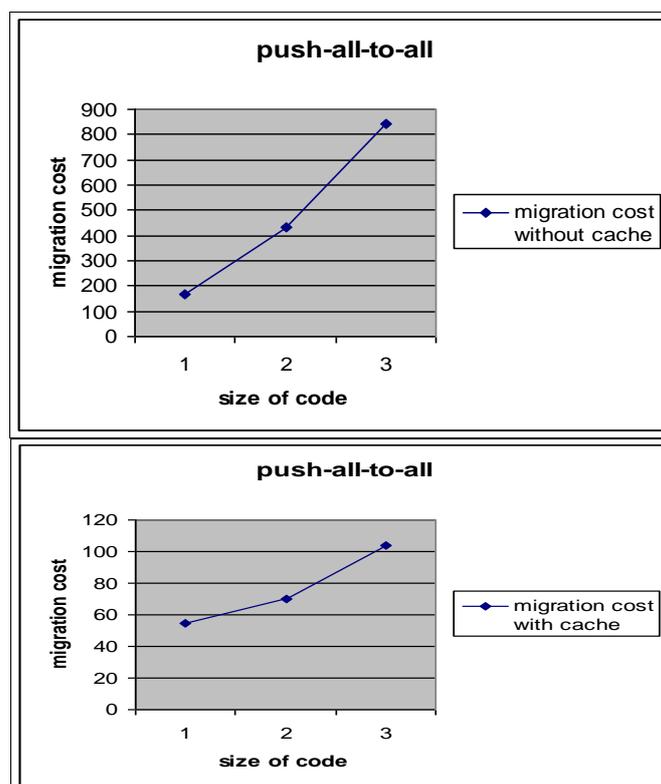


Fig6: Push all to all

V. CONCLUSION

The migration cost of mobile agent is calculated by adding the freezing time of all the nodes or sites. From above table when the size of MA code is varied from 7.5K to 160K, the freezing time of caching-added scheme is lower than the freezing time of without caching scheme. As the size of MA code increases, the performance of the scheme is improved from 118% to 125%.

REFERENCES

- [1] Gilbert, D. Aparicio, M. Atkinson, B. Brady, S. Ciccarino, J. Grosz, B. O'Connor, P. Osisek, D. Pritko, S. Spagna R. and Wilson, L., "IBM Intelligent Agent Strategy", IBM Corporation, 1995.
- [2] Alfonso, F., Gian Pietro, P., and Giovanni, V., "Understanding Code Mobility", IEEE Transactions on Software Engineering, vol. 24, No. 5, 1998
- [3] Danny B. Lange and Mitsuru Oshima, "Seven Good Reasons for Mobile Agents", Communications of ACM, vol. 42, No. 3, 1999
- [4] LIU Da-you, Yang Bo "Strategically migrating agents in itinerary graph" Proc. Of second international conference on machine learning and cybernetics, vol-3, 2003, pp-1871-1876
- [5] Peter Braun, Christian Erfurth "Performance Evaluation of various migration Strategies for Mobile Agent" 2001, LNCS 1998, pp. 1-12.
- [6] Susmit Bagchi, "Ubiquitous Multimedia And Mobile Agents: Models And Implementations", IGI Global, 2012. First Edition.
- [7] Christian Erfurth, Wilhelm Rossak "Autonomous itinerary planning for mobile Agent" Proc on the third Symposium on Adaptive Network, 2003, pp-940-945
- [8] LIU Da-you, Yang Bo "Strategically migrating agents in itinerary graph" Proc. Of second international conference on machine learning and cybernetics, vol-3, 2003, pp-1871-1876