Improved Mobile Agent Migration In A Heterogeneous Environment Via Reverse Migration

Aditya Vikram Bhunja
Final Year Student,
Bachelor of Computer Applications
Institute of Engineering & Management, Kolkata

Abstract

This paper basically proposes a way to reduce the network load in the Cyber Net while following a Agent Regeneration technique but having a reverse migration path for the Mobile Agent.

1. Introduction

In computer science, Agents (synonymously called Software Agents) are computer programs which resides and stays attached in the system itself to perform tasks on behalf of the user. So as in definition of Mobile Agents we can say that they are Software Agents with the gift of mobility. This means that it can migrate from system to system with its execution code and status. The main advantage of Mobile Agent technique over the traditional technique is the reduction of network load.

In a distributed system, for completing a task there are multiples number of times information will moves form one node to another node. This is especially true when security measures are enabled. So that the result is a lot of network traffic and size of data will increase because, security thread will add every time when data is sent. Mobile agents allow us to dispatch all information of the task at a time when agent arrives at destination host the interactions can take place locally rather than transferred over the network. The objective of mobile agent bases communication is to move the computations to the data rather than the data to the computations.

This paper now stresses on further network load reduction in spite of usage of mobile agents.

2. The Problem

2.1. Problem Statement

As stated earlier, the main advantage of Mobile Agent technique over the traditional technique is the reduction of network load. In other words, the existence of Mobile Agents has itself relieved the Cyber Net of a lot of

network burden. This paper stresses on further network load reduction in spite of usage of mobile agents.

2.2. Problem Description

The problem consists of a minimum of three heterogeneous machines or environments through which the mobile agent migrates. Namely we take the three environments as E1, E2 and E3. Here E1 is the home machine and E2, E3 are the host machines. Now the normal mode of migration is shown in Figure 1 and reverse mode migration is shown in Figure 2.

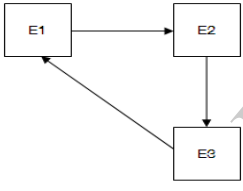

Figure 1. Normal Agent Migration

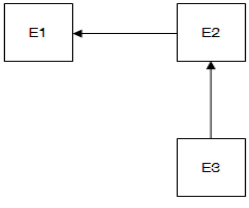

Figure 2. Reverse Agent Migration

As we can clearly see and compare the two figures, we see that one agent migration is lesser in case of Figure 3 than in case of Figure 2 i.e. from E3 to E1. Now this lowers the network load considerably by following reverse migration.

3. The Solution

3.1. Solution Assumptions

- All RPC calls made by one node to other nodes are secure.
- All nodes in the network framework are active, trusted and non-malicious.
- The migrating mobile agent is non-malicious.

3.2. Solution Components

- E1, E2, E3- Heterogeneous environments.
- Code Server- Where the codes of the mobile agent of home machine are stored.
- Agent Blueprint- It is the skeletal structure of the programming code and functionalities of the mobile agent to be sent. In other words, it is an implement-tation independent descript-tion of the agent.
- Class Loader- It is a piece of software which creates an empty mobile agent using the blueprint and is also responsible of loading codes (downloaded from the code

server) into the empty agent.

- Authenticator- It is a token that is sent along with the mobile agent so they at least can enter into the architecture of the host machine.
- Forward Pointer Scheming- It is a procedure of the home machine to send the host machines the node address of the agent's next hop according to the routing table.
- Backward Pointer Scheming- It is a procedure of the home machine to send the host machines the node address of the agent's previous hop according to the routing table.

3.3. Solution Description

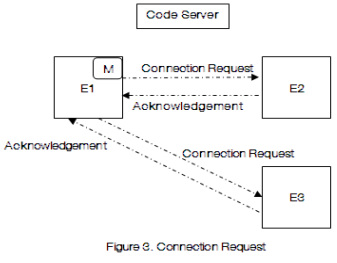

Figure 3. Connection Request

Firstly a mobile agent is created at the home machine. Then the home machine makes connection request to the host machines. After receiving acknowledgment from the host machines (as shown in Figure 4) in a stipulated time a routing table is made in the home machine and passed to the code server.

Now the made *agent's information* is sent to the host machines. The *agent information* contains the following details about the mobile agent:

- Agent's blueprint
- Home machine's authenticator.
- Address of next hop (forward pointer scheming).
- Address of previous hop (backward pointer scheming).

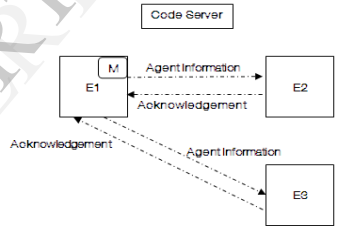

Figure 4. Passing Agent Information

After receiving the information, the hosts again send back an acknowledgement. This procedure is diagrammatically described in Figure 5.

Now a skeletal agent (with no mobile agent) is now created in E3. In this case, the newly-made agent is embedded with the following information within itself:

a) Home machine's address
b) Home machine's authenticator.

c) Destination host machine's address (E2's address).
d) Previous hop address (also Home Machine's Address).
e) Data.

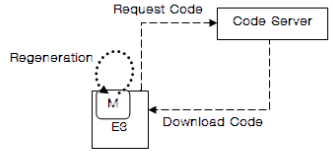

Figure 5. Agent Regeneration in E3.

Now a request for the mobile agent code is done to the code server of the home machine E1. The code server confirms the authenticator with the home machine via RPC, if the confirmation is positive then the request is granted else not. If the request is granted then a new mobile agent is re-created or regenerated within the host machine according to its configuration via the Class Loader. This whole process is well described in Figure 5.

After execution, the new mobile agent is cloned and dispatched from the current environment while appending the next hop address in it. After it has been dispatched, a signal is sent to the home machine of its dispatch. The home machine then sends an auto-destruct command to both the sent mobile agent and the new mobile agent which still resides in E3. The new mobile contains the following components:

a) New updated data.
b) Home machine's address
c) Home machine's authenticator.
d) Destination host machine's address (E2's address).
e) Previous hop address (E3's Address).

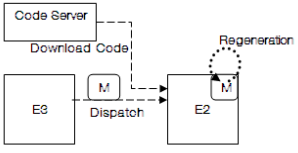

Figure 6. Migration from E3 to E2

On reaching E2, as shown in Figure 6, the same methodologies are used to tackle the execution and dispatch except all data in the received agent i.e. from E3, is extracted before execution.

On reaching E1, the environment again regenerates the agent (as shown in Figure 7) and destroys the received agent after extracting its data.

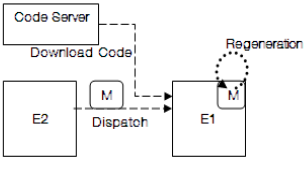

Figure 7. Migration from E2 to E1

4. Drawbacks

- The migration dependency increases on the last node environment since it initiates the migration.

- Regeneration is done even if both environments are homogenous in nature.

- All security issues like eavesdropping, masquerading, etc. are not covered.

5. Conclusion

The proposed framework not only supports heterogeneous mobility, offering an agent maximum flexibility with respect to where it wants to go, but also reduces the network load to a certain extent. As a result, this can be a good solution to deal with the heavy traffic in Cyber Net.

6. References

[1] Aditya Vikram Bhunja, Optimized Mobile Agent Migration Overcoming a Unique Deadlock Scenario in a Heterogeneous Environment, International Journal of Engineering Research and Technology, Volume 2 Issue 4, April 2013, pp. 2320-2326.

[2] D. B. Lange and M. Oshima, Seven good reasons for mobile agents, Communications of the ACM, 42 (1999), pp. 88–89.

[3] L. Magnin, T. V. Pham, A. Dury, N. Besson, and A. Thiefaine, Our guest agents are welcome to your agent platforms, in Proceedings of the ACM Symposium on Applied Computing (SAC 2002), Madrid, Spain, Mar. 2002, pp. 107–114.

[4] D. C. Smith, A. Cypher and J. Spohrer (1994) "Programming Agents without a programming language" Communications of the ACM 37 (7) pp 55-67.

[5] J. White. "Mobile Agents".In J. M. Bradshaw (editor), Software Agents. AAAI Press/MIT Press, 1997, pp. 437-472.

[6] P. Morreale. "Agents on the move". In IEEE Spectrum, April 1998, pp. 34-41.

[7] B.J. Overreinder, D.R.A. De Groot, N.J.E. Wijngaards and F.M.T. Brazier (2006), "Generative Mobile Agent Migration in Heterogeneous Environments" in Scalable Computing: Practice and Experience, volume 7, number 4, pages 89-99.

[8] Xinyu Feng (2002), "Design and Analysis of Mobile Agent Communication Protocols", Institute of Computer Software Nanjing University, China