A tutorial report for SENG 609.22
Agent Based Software Engineering
Course Instructor: Dr. Behrouz H. Far

Mobile Agents

Samuel Lee

Department of Electrical Engineering
University of Calgary

Abstract

With the increasing popularity of internet, and distributed heterogeneous network, there comes a new technology innovation that lead a system to be based on some form of mobile codes especially mobile agents. In this paper, the topic of mobile agents and some of the benefits and challenges of this new technology will be discussed.

Introduction

Rapid development and new innovations in the computer and network technology had led the revolutionary advancement in the information world. The amount of information and services that is available through the Internet, television, pagers, cell phones from anywhere and everywhere are overwhelming. Mobile code and especially mobile agents is extremely important in the digital world today because it aids the flexibility of accessing the information from any location.

Mobile agents are autonomous, intelligent software programs which meant that they can decide where they will go and what they will do. They can migrate from host to host in a

network, searching for and interacting with services to complete a specific task, which means that mobile agents are also goal oriented. One of the key features of the mobile agents is that not only the code is mobile but the state of the agent is also mobile. This means that mobile agents should be able to be understood by every machine in the network without having to install the agent codes every where and also be able to carry the incomplete transaction or calculation with them when they migrate. There must also be a server environment that will accept these agents and allows the agents to do certain processes or making certain requests to this server environment. With the widespread adoption of the java, it provides the environment that the mobile agents needed and hence it also increases people's interests in mobile agents.

Mobile Agent Architecture

Mobile Agent's Movement

Before we discuss the mobile agent architecture let's take a look at the traditional approach of the client/server network, this will help us to understand the migration behavior of mobile agents.

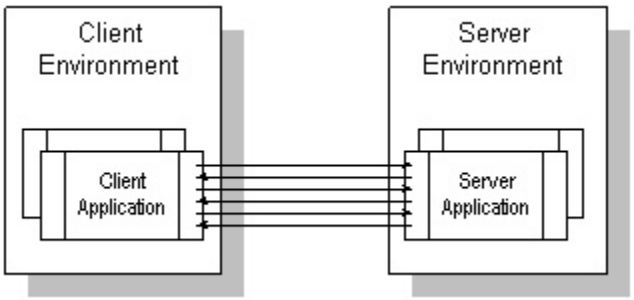

Figure 1: Typical client/server application communicates via requests and responses [1]

Figure 1 shows the typical client / server communication where in most cases, there are a client side and a server side application where these applications are located at different machines. When ever the client requires certain resources or data from the server, a request is sent to the server and then server will acknowledge this request and send a response back to the client. The client will then start sending data over to the server and this type of communication will be continued until a request is completed. Each time a request send from the client and the response send back to the client from the server, the data packets will need to do a complete round trip around the network, this type of communication consumes a lot of bandwidth and it is also not very time efficient.

Now let's take a look at the mobile agent architecture.

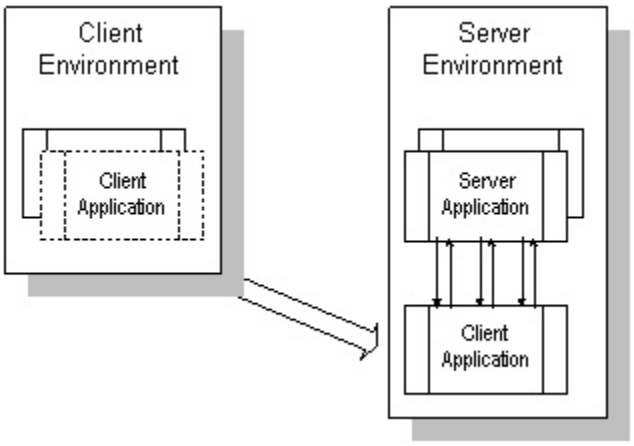

Figure 2: mobile agent architecture [1]

In this architecture, the agent on the client machine actually migrates to the server machine, and it will not only carry the code but also the state of the agent. The agent does not talk to the server across the network, it actually moves itself over to the server and then the agent will make any request to the server directly. Once it is completed, the agent will then move itself back to the client side carrying the results. One thing that is worth mention is that not only mobile agents can migrate from client to the server but it can also move from server to server and it will drag the incomplete transaction or calculation with them until it is finished and then it will return home. The actual term for the computing location where the agents migrate is called the hosts. These hosts provide the environment for mobile agent to work and also are responsible for packaging and moving these mobile agents.

Communication method between agent hosts

In order to transport the mobile agents between the hosts there must be some form of communication between the hosts. Currently, there are no requirements for hosts to use a special type of communication protocol. Hosts can use TCP/IP or other higher level communication protocol such as SMTP (simple mail transfer protocol), RMI (remote method invocation), IIOP (Internet Inter-ORB Protocol) or HTTP (hypertext transfer protocol). Mobile agents can use variety of transporting mechanism which gives them the flexibility.

Mobile agent lifecycle

Agents have a well-defined lifecycle [1]. Figure 3 below illustrates the four states of the mobile agents. At the beginning, there will be an initialization of the agent. The agent will then start to function and may stop and start again depending on the environment and the tasks that is tried to accomplish. After the agent finished all the tasks that are required, it will end at the completing state.

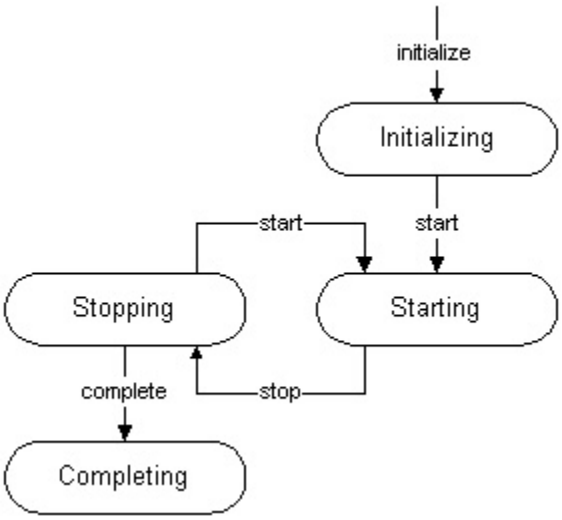

Figure 3: lifecycle of the agent [1]

Benefits of the mobile agents

So why do we have to use mobile agents? What are the benefits of these mobile agents? Here are some the reasons:

Bandwidth. Even though there are astonishing amount of fiber being installed around the world. But one still can not deny that there are still a large number of end users that are still using 56K modems and ADSL over the old copper loop. My company had done a recent study on some of our clients, and we had discovered that many of the employees of big corporations are still using modems. Even some of the remote offices of big corporations, there internet connect is no better than 56K modems. Therefore, the bandwidth to many end users is still a scarce resource. The mobile agents can conserve the bandwidth by migrating itself to the hosts and after all the process is finished it will then return home with the result. This eliminates that hundred or thousand of queries that had to go across the network. Again, this is based on that the amount of queries that traveled back and forth would consume more bandwidth than the single large burst of information (agent) traveling from client to the server and back.

Mobile users & mobile devices users. There are more people using the PDA, cell phones, laptops, etc. These devices either do not have lots of computing powers or these devices have unreliable connections. By moving the processing to the server side will help save a lot of the client resources. Another benefit of the mobile agent is that even though the remote user loose connection on their end, the mobile agent will still work hard on the hosts' site to finish the task and will return to the client once the client is connected again.

Communication Latency. Communication latency can be greatly reduced. Since agent carries the entire service request with them across the network, there is no need

to send the queries or requests back and forth between the server and the client. This in term saves the time to complete the task.

Fault Tolerance. Since the communication between the client and the server is not ongoing. Therefore, the mobile agents are not as susceptible to faults as the traditional client-server architecture, since there is less dependency on the network and the communication between client and server.

Distributed Heterogeneous Computing. Mobile agents can provide a single, general framework where distributed applications can have loads distributed evenly across the whole information infrastructure. For example, agents can collect data on one node and transfer to another.

Scalability. Hosts can be added easily to mobile agents.

Challenges

In this section, we will explore some of the technical and non-technical challenges that the mobile agents will face.

Security. There are few challenges on how mobile agent system can protect the resources against malicious agents from the following:

1. Authentication of the sender of the mobile agent, and authentication of the server or agent in the execution environment.
2. Authorization for the agents and hosts. There must be some kind of rule or regulation in order to prevent the agent from infecting the hosts, deny service to other agents or infecting other agents.
3. Limiting the number of resources that agents can access. How do we prevent an agent from becoming resource hog? Once the agent is inside the host, we must protect the available resources on the host and still allowing the agent acts autonomously.

There must be a balance between protecting the hosts and also allowing the agents to have the access rights to perform their tasks.

Standardization. There are many mobile agent frameworks that exist today. There must be standardization on some specific execution environments and the format on how mobile agents should be encoded in term of the code and state, which allows mobile agents to work with other mobile agents.

Lack of applications. The mobile agents in most aspects are considered to be new programming paradigm. There are not many applications out there that use mobile agents so it is hard for this technology to become widely adopted. Most work had been done around the agent's framework instead of developing the real applications.

Competing Technologies

As of today, mobile agents still facing tough competition from other technologies that can achieve similar goals or tasks as mobile agents. In this section, we will discuss two popular alternatives to mobile agents, the message passing system and RPC (remote procedure call). In both methods, data are passed without specifying on how the data are to be processed; each has implicit knowledge of the capabilities of the remote procedures [3].

Message passing system

It is an outgrowth of both electronic mail systems and earlier distributed computing schemes in which applications communicated via files or pipes [4]. In this message passing system, the client will compose a message where usually is a structured text, and then this message is delivered to an appropriate software processor for the type of the message. One example of the message passing system is the KQML (Knowledge Query Manipulation Language). It allows agents to communicate using performatives (a rich set of messages), but it does not require mobility. KQML can simply pass a message and then deliver this message through the transport mechanism. [3]

RPC (remote procedure call)

It extends the traditional procedure call mechanism of pushing parameters, registers, and a return address onto the stack and then performing a jump to the procedure's entry point [3]. RPC is synchronous and it behaves like the local procedure call where the client will suspend its process until the return RPC is received from the server.

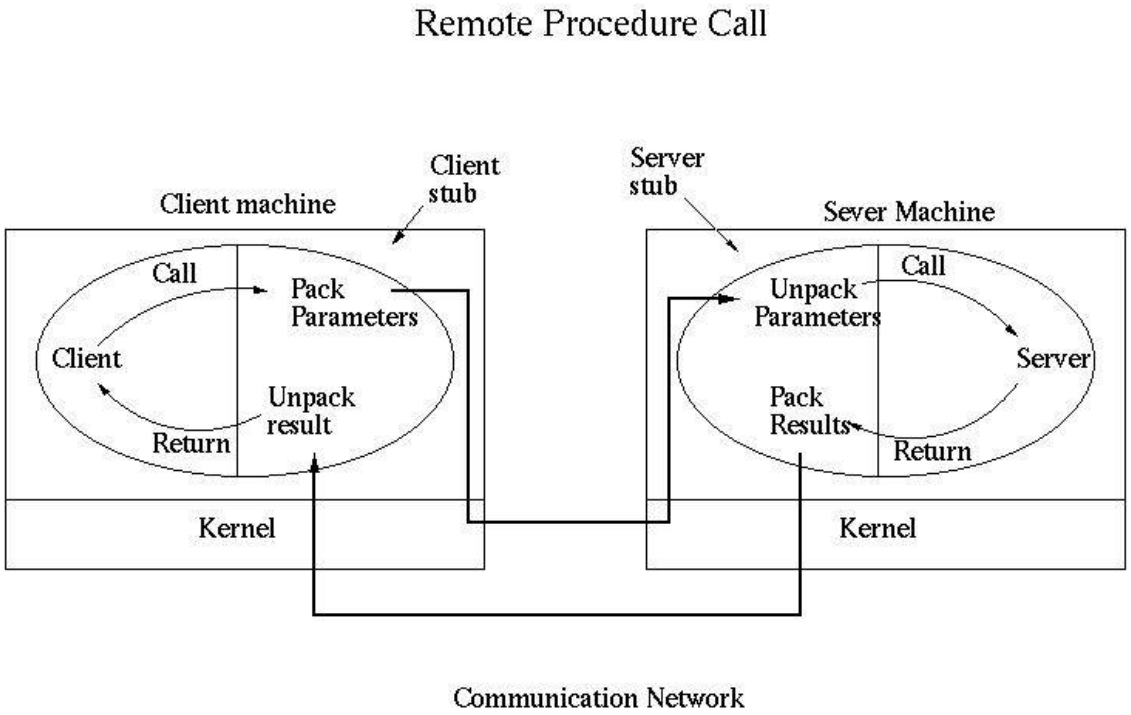

Figure 4: RPC [5]

Commercial Mobile Agent Systems

Even though mobile agents are relative new, there are a lot of commercial mobile-agent systems that exist on the market today. In this section, we will compare some of the most popular commercial mobile agent systems.

Mobile agent systems

Mobile Agent System	Author	Language	Secure Communication	Server Resource	Agent Protection
Telescript	General Magic	Created their own OO, type-safe language	Agent transfer is authenticated using RSA and encrypted using RC4	Capability-based resource access. Quotas can be imposed. Authorization based on agent's identity	Not supported
Tacoma	Cornell University University of Tromso, Norway	Tcl, but is created to be written in other scripting languages	Not supported	Not supported	Not supported
D'Agents	Dartmouth College	Tcl interpreter, modified to execute scripts and capture state of execution at thread level	Uses PGP for authentication and encryption	Uses safe-Tcl as its secure execution envireon ment. No support for owner-based authorization	Not supported
Aglets	IBM	Java. IBM developed a separate class library to create mobile agents	Not supported	Statically specified access rights, based on only two security categories: trusted and untrusted	Not supported
Voyager	ObjectSpace	Java. Unique feature is a utility which takes any Java class and creates a remotely-accessible version of it.	Not supported	Programmer must extend Security Manager. Only two security categories: native and foreign.	Not supported
Concordia	Mitsubishi Electric	Java. Has Itinerary object, which keeps track of an agent's migration path	Agent transfer is encrypted and authenticated using SSL	SecurityManager screen acceses using a statically configured ACL based on agent owner identity	Agents protected from other agents via the resource access mechanism
Ajanta	University of Minnesota	Java	Transfer is encrypted using DES and authenticated using ElGamal protocol	Capability-based resource access. Authorization based on agent's owner	Mechanisms to detect tampering of agent's state and code

Figure 5: Mobile Agent Systems [6]

Conclusions

Mobile agent is the new program paradigm that is still searching for the widespread of acceptance. Even though there are few obstacles such as the security and standardization but with further research, these obstacles can be overcome. I think the benefits of mobile agents will make an attractive technology choice for the internet industries, especially with the increasing number of mobile users and the new advancement on mobile devices.

REFERENCES

[1] Todd Sundsted, "Agents on the move", http://www.javaworld.com/javaworld/jw-07-1998/jw-07-howto_p.html, July 1998.

[2] David Kotz and Robert S. Gray, "Mobile Agents and the Future of the Internet", http://www.cs.dartmouth.edu/~dfk/papers/kotz:future2.pdf, 1999

[3] David Reilly, "Mobile Agents – Process migration and its implications", http://www.davidreilly.com/topics/software_agents/mobile_agents/, 1998

[4] Colin Harrison, David Chess, Aaron Kershenbaum, "Mobile Agents: Are they a good idea?", IBM Research Division, 1995

[5] Author unknown, "Distributed Computing – II", http://www.ncst.ernet.in/~ppfac/slides/distrib-comp.ppt, 1997

[6] Olga Gelbart, "Mobile Agents", http://www.seas.gwu.edu/~simhaweb/ecom/agents.ppt

[7] Joseph Kiniry, Daniel Zimmerman, "A Hands-On Look at Java Mobile Agents", http://computer.org/internet, August 1997