Mobile Agents- Theoretical Approach

M.N. Thippeswamy *

Abstract

Mobile agents are agents that can physically travel across a network, and perform tasks on machines that provide agent hosting capability. This allows processes to migrate from computer to computer, for processes to split into multiple instances that execute on different machines, and to return to their point of origin. Unlike remote procedure calls, where a process invokes procedures of a remote host, process migration allows executable code to travel and interact with databases, file systems, information services and other agents. The technology behind mobile agents is examined, and an analysis of its uses and implications is offered. Use of the Internet has exploded in recent years with the appearance of the World-Wide Web. In this paper, how current technological trends may lead to a system based substantially on mobile code, and in many cases, mobile agents it seems likely that, within a few years, nearly all major Internet sites will be capable of hosting and willing to host some form of mobile code or mobile agents are theoretically discussed.

Keywords

Remote Procedure Call (RPC), Transmission Control Protocol / Internet Protocol (TCP/IP), Remote Method Invocation (RMI), Hyper text Transfer Protocol (HTTP), Simple mail Transfer Protocol (SMTP), Central processing Unit (CPU), Common Object Request Broker Architecture (CORBA), Knowledge Query Manipulation Language (KQML), Internet Service Provider (ISP), Wide area Networks (WAN), Application Programming Interface (API)

1 Overview

Mobile agents have been the focus of much speculation and hype in recent years. The appeal of mobile agents is quite alluring - mobile agents roaming the Internet could search for information, find us great deals on goods and services, and interact with other agents that also roam networks (and meet in a gathering place) or remain bound to a particular machine. Significant research and development into mobile agency has been conducted in recent years, and there are many mobile agent architectures available today. However, mobile agency has failed to become a sweeping force of change, and now faces competition in the form of message passing and remote procedure call (RPC) technologies.

1.1 Definitions

Proxies: Such proxy sites, which today are most often Web sites interpose between a user and one or more other Internet services. As a means to both reduce information overload and customize service access, proxy sites will become more and more important. In particular, as portable devices become more prevalent, highly specialized proxy sites will be provided to meet the special needs of mobile users.

Internets: Organizations are increasingly using Internet protocols, particularly HTTP, to build internal "intranets" for their own distributed-information needs. All access to an intranet is managed by a single organization. Thus, new technologies can be deployed quickly, since (1) little coordination is needed with outside organizations, and (2) security (within the internet) is of less concern.

2 Technology behind process migration

In general, the following things are required to allow agents to migrate across a network

1. Common execution language
2. Process persistence
3. Communication mechanism between agent hosts
4. Security to protect agents and agent hosts

2.1 Common execution language

If a process is to migrate from one host to another, then both hosts must share a common execution language. In a homogenous networking environment, it's conceivable that assembly language or machine code could be sent across the network for execution. However, such a system would be extremely limited and not very future proof.

*M.N. Thippeswamy Lecturer, Electrical Engineering Dept, Faculty of Engineering, Bahir Dar University

A more likely scenario for mobile agency is a heterogeneous environment, where many different system architectures are connected. In this case, an interpreted scripting language or emulation of a system that is capable of executing machine code solves the problem of a common execution language.

2.2 Process persistence

For processes to migrate to remote machines, they must be capable of saving their execution state, or spawning a new process whose execution state will be saved. This property is called persistence. Persistence involves converting the object's state (variables, stack, and possibly even the point of execution) and converting it into a data form suitable for transmission over a network. Agents should not have to be responsible for achieving this themselves, and process persistence would likely be built into the mobile agent language or architecture.

2.3 Communication mechanism between agent hosts

Some communication mechanism must exist to transfer agents across networks. An agent might be transferred using TCP/IP, or by using a higher level of communication such as RMI, SMTP or even HTTP. Mobile agent architectures may even use a variety of transport mechanisms, giving greater flexibility.

An agent's executable code must be transferred, which may consume a large amount of network bandwidth, unless shared code is located at the agent host. Techniques such as shared libraries of code, or caching, may be of benefit. In addition, the persistent state of the agent must be transferred.

2.4 Security to protect agents and agent hosts

Security is critical when executable code is transferred across a network. Malicious or badly written code could wreak havoc when unleashed upon an unsuspecting host, and agents themselves need protection against hostile hosts that would seek to dissect or modify them. There is no magic solution that will solve all the security problems of mobile agents, but precautions can be taken to minimize risk.

When an agent leaves for a new host, extreme care must be taken to prevent unauthorized modification or analysis of the agent. Agents may carry with them confidential or sensitive information and logic, which shouldn't be accessible to the agent host. Encryption may be of benefit, but the data and code must be decrypted at some point in time for the agent to execute. Once this occurs, the agent becomes vulnerable, and is at the mercy of the agent host. In a scripting language, the internal logic of the agent is exposed, but even compiled languages can be decompiled with a disturbing degree of success. Other than using trusted hosts, there is little that can be done to protect the agent from snooping eyes.

For agent hosts, the news is a little more positive. Through the use of digital signatures, the identity of an agent and its user can be authenticated. However, in many commercial environments, even un-trusted agents will need to be accepted. These agents may carry with them dangerous payloads, such as code that needlessly consumes CPU time and memory that attempts to damage the agent host or carries back information that would make it vulnerable to hacking. Such is the risk an agent host must take, though limiting access to resources such as disks and the network may offer some solution.

3 Implications of mobile agents

If mobile agents were to gain widespread commercial adoption (to the degree that say, web browsing or email has), then what type of a network would we have? The following list of implications is by no means exhaustive, but does provide an interesting set of observations.

3.1 Bandwidth conservation

One of the goals of mobile agency is to conserve bandwidth, by placing an agent directly at the point of information, rather than sending dozens or even hundreds of queries across the network (General Magic, 1998). This is based on the premise that these queries would have consumed more bandwidth than sending an agent over the network, and bringing it back again.

Bandwidth conservation is an admirable goal, but whether mobile agents will help realize this goal is questionable (Harrison, 1995). For bandwidth to be conserved, the bandwidth consumed by sending across a mobile agent, and waiting for its results, must be less than that of a series of queries sent via a messaging or RPC system. This is a determination that must be made in practice, and cannot be fully verified just by theory.

Many scenarios can be foreseen. Perhaps mobile agents could comb through large amounts of resources on a single site, and bring back a small number of matches, in a similar nature to a search engine. However, some electronic commerce models suggest that mobile agents would be sent out to multiple sites, perhaps to negotiate low prices with vendors. This sort of activity has the potential to result in an incredible explosion of bandwidth consumption.

Indeed, searching is a task that many people would like to see mobile agents performing. Currently, a small number of indexing agents collect information for search engines, while millions of queries are made by users. Imagine if the same number of queries were made instead by mobile

agents that traveled across the network to sites. Two scenarios are possible. Either a much larger amount of bandwidth (and CPU usage for the agent hosts) will be consumed, or a much lesser amount of bandwidth will be consumed as users receive more accurate search results because their agents have more control over the search process. Instinct suggests, however, that a simple keyword query entered via a web browser will consume less resources and bandwidth than sending an agent with specialized searching algorithms across the network.

3.2 Delegate tasks to agents when not connected

The Internet, as it stands today, is made up of many millions of computers, some of which are permanently connected but the majority of which connect via dial-up modem connections for short periods of times. Imagine if you could delegate tasks to mobile agents that would roam the network for you while not connected. This goal would be extremely desirable in the short term, until permanent connections became more prevalent. Here mobile agents may have found a sound market. This market could potentially be profitable, for the mobile agent technology vendors and the agent hosts that allow offline usage of their network.

Delegation of tasks to mobile agents could also be used as a form of load sharing in distributed systems. Agents could perform tasks on remote systems, moving from system to system as required to balance the load. Mobile agency also gives greater flexibility, because new tasks and new code can be added to the system without the need for a fixed database.

3.3 Mobile agents enable new types of interaction

The ability of mobile agents to fragment themselves into many pieces that travel to different points across the network sounds promising. It might enable new forms of interaction, such as negotiating agents that travel to vendors seeking the best deal, or meeting places where agents can "get together" and communicate. The attraction of mobile agents for electronic commerce is great, and it might make sense to deploy mobile agents for electronic commerce. However, such uses could also be accomplished by message passing, or direct communication using application protocols like HTTP. Mobile agency is promising, but it is not the only mechanism for new uses of software agents.

3.4 Agent privacy

If mobile agents were to become common place, serious privacy concerns would be raised. Aside from deliberate attempts to decompile or interrogate an agent (or its encrypted data), agent hosts could also monitor the actions of agents, and create consumer profiles. Even knowledge of the types of queries, or the way in which an agent searched, could reveal information about its owner. When individuals query a search engine, there is some degree of anonymity, but there is less control with mobile agency.

3.5 Competing technologies

Mobile agency faces stiff competition from other technologies that can achieve similar outcomes, such as message passing or advanced forms of remote procedure calls. Some of the more notable technologies that offer competition to mobile agency are discussed.

3.5.1 Message passing systems

Software agents need not always travel across a network to communicate with information sources, or other agents. One of the most important message passing systems for agents is the Knowledge Query Manipulation Language (KQML). KQML is an important mechanism for communication, because it allows much more complex forms of interaction than query/response mechanisms. KQML allows agents to communicate using a rich set of messages called performatives, and is capable of communicating attitudes about information, rather than just data and facts (Finin, 1998). Message passing systems like KQML don't require mobility; they can simply pass a message and have it delivered through some transport mechanism.

3.5.2 Remote Method Invocation (RMI)

Remote method invocation allows Java developers to write distributed systems that share objects. New objects can be transferred across the network, and RMI is becoming a popular mechanism for agent communication. RMI can be used to facilitate mobile agency (acting as a transport mechanism), or as a replacement that allows agents to invoke methods of other agents.

3.5.3 Common Object Request Broker Architecture (CORBA)

CORBA is a platform and language independent mechanism for invoking remote object methods. Unlike RMI which is specific only to Java Virtual Machines, CORBA can be used to create distributed systems that execute on many platforms, in many languages. CORBA holds great potential, because of its portability and flexibility. CORBA is a direct threat to mobile agency, and would allow developers to create agents that are capable of complex communication without ever traveling across a network.

4 Mobile agents are coming as part of the Internet

The trends outlined in the previous sections lead to the conclusion that mobile code, and mobile agents, will be a critical near-term part of the Internet. Why? Not because mobile code makes new applications possible, nor because it leads to dramatically better performance than (combinations of) traditional techniques, but rather because it provides a single, general framework in which distributed, information-oriented applications can be implemented efficiently and easily, with the programming burden spread evenly across information, middleware, and client providers. In other words, mobile code gives providers the time and flexibility to provide their users with more useful applications, each with more useful features.

The argument roughly follows Figure 1.

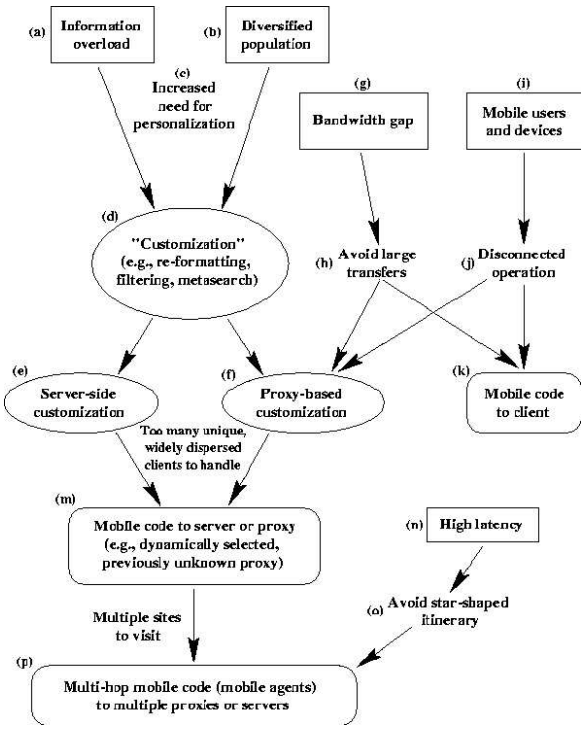

Figure 1: The trends leading to mobile agents

Both the amount of information available on the Internet (a), and the number and diversity of its users (b), are growing rapidly. This diverse population of users will not settle for a uniform interface to the information, but will demand personalized presentations and access methods (c). This personalization will range from different presentation formats to complex techniques for searching, filtering and organizing the vast quantities of information (d). Today, such personalization facilities are provided at the information source in a site-specific manner (e), at a proxy Web site (f), or (occasionally) as client software.

Meanwhile, the network technology will lead to an increased gap in the bandwidth of the core Internet versus the fringes of the Internet (g). Thus, most client hosts will shun large transfers of data (h). That trend encourages the migration of application functionality from clients into proxy sites (f), which are presumably better connected to the core Internet, and need send only the final results over the slower connection to the client. Furthermore, the dramatic availability of core bandwidth will allow these proxy sites to be aggressive in gathering, prefetching, and caching information on behalf of their clients.

Mobile users (i) will frequently disconnect from the network, and perhaps connect later at another location with poor bandwidth (j). This tendency again leads to the use of proxies (f). It also encourages application programmers to choose a mobile-code solution to dynamically install the necessary client code (k) onto the Web terminal or portable device. Moving code (applets) to the client allows a high level of interaction with the user despite a high-latency, low-bandwidth, or disconnected network.

Ultimately, Web sites and other Internet services will not be able to efficiently provide the full range of customization desired by their clients, and clients will want to use the same information-filtering and -organizing tools across many sites. Moreover, fixed-location, application-specific proxies will become bottlenecks, and as user needs change, may no longer be at the best network location for accessing the proxies services. As a result, customization tools will be specified as software, in the form of mobile code that runs either on the server, or on a dynamically selected proxy site (m) near the server (or client, as appropriate). Mobile code is necessary, rather than client-side code, since many customization features (such as information monitoring) do not work if the client is disconnected, has a low-bandwidth connection, or requires frequent communication with the server. Mobile code is beneficial, since servers and proxy sites need provide only a generic execution environment (along with an API that provides programmatic access to their service); the actual customization tools can be written by the services themselves, by third-party middleware developers, and even by the end users.

Finally, many clients will wish to send mobile code to multiple information sites as part of a single task. Although there will be applications for which the mobile code can be sent in parallel, many tasks require a sequence of subtasks, each at a different site. To avoid latency (n), the application programmer will often want to avoid a “star-shaped graph” (o) where mobile code goes out to the first site and sends its results back to the client or proxy, the same or different piece of mobile code goes out to the second site, and so on, and the programmer will always want to be able to select the best migration strategy for the task and current network

conditions. In other words, the mobile code must be able to hop sequentially through multiple sites; such multi-hop mobile code is a mobile agent.

4.1 The widespread adoption of mobile-agent technology

Internet sites must have a strong motivation to overcome inertia, justify the cost of upgrading their systems, and adopt the technology. While the technological arguments above are convincing, they are not sufficient for most site administrators. In the end, the technology will be installed only if it provides substantial improvements to the end-user's experience: more useful applications, each with fast access to information, support for disconnected operation, and other important features.

4.1.1 Getting ahead of the evolutionary path

It is unlikely that any Internet service will be willing to jump directly from existing client-server systems to full mobile-agent systems. Researchers must provide a clear evolutionary path from current systems to mobile-agent systems. In particular, although full mobile-agent systems involve all the same research issues (and more) as more restricted mobile-code systems, researchers must be careful to demonstrate that the switch to mobile agents can be made incrementally.

4.1.2 Recommended idea for Implementation

For example, " applets", mobile code that migrates from server to client for better interaction with the user, are in common use, and the associated commercial technology is improving rapidly (e.g., faster Java virtual machines with just-in-time compilation). From applets, the next step is proxy sites that accept mobile code sent from a mobile client. In all likelihood, such proxies will be first provided by existing Internet service providers (ISPs). Since the sole function of the proxy sites will be to host mobile code, and since the ISPs will receive direct payment for the proxy service (in the form of user subscriptions, although not likely at a fixed rate), the ISPs will be willing to accept the perceived security risks of mobile code. Once mobile-code security is further tested on proxy sites, the services themselves will start to accept "servlets", mobile code sent from the client directly to the server (or from the proxy to the server). Once servlets become widely used, and as research address the issue of protecting mobile code from malicious servers, services will start to accept mobile agents.

4.1.3 Another critical evolutionary path

Is the migration of agent technology from intranets to the Internet. Mobile-code technologies will appear first in the relatively safe intranet environment, particularly intranets that are built on high-latency networks such as a WAN or a wireless network for mobile computers. For example, a large company, particularly one with a mobile workforce, might find mobile agents the most convenient way to provide its employees with a wide range of access to its internal databases. Intranets tend to be early adopters of new (useful) technology, because their administrators have more control over the intranet than over the Internet; that control means that security is less of a concern, and wide deployment of agent support services can be encouraged. As the technologies mature in intranets, site administrators will become comfortable with them, and their practicality, safety and potential uses will become clear. Then they will find their way into the Internet.

5 Examples

Various mobile application running on mobile agents system have been developed so far, for example work flow management,CSCW, distributed information retrieved, active networks, and so on. one of the important examples is applications based on the concept of compound documents like OpenDoc developed by Apple Computer and IBM.This application allows compound documents given as mobile agents to be dynamically composed into a compound document. as shown in figure 2.

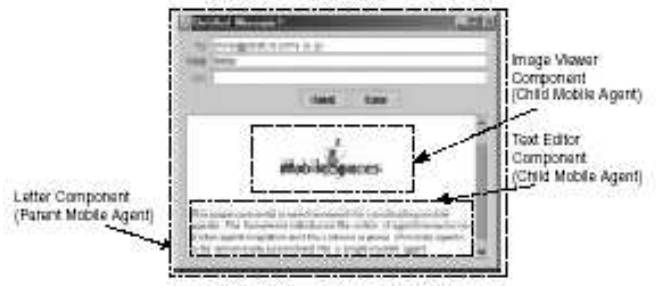

Figure 2: Window of the Compound Letter Agent

Another an example is An electronic mail system where each letter is a mobile agent incorporated with the frame work as shown in figure 3.Therefore, each letter can contain more than one mobile agent-based component:some text, graphics, and animations on the document of the letter as shown in figure2 and figure 3. Users can edit these inner components written in arbitrary data formats, because they are mobile agents and thus can include programs to edit their own components. For example, to edit the text, simply click on it.and its editor program is invoked. The letter agent can autonomously deliver itself and its inner components to the destination. The receiver can read all the contents of the

arriving letter, because the letter is a mobile agent that contains its components to view the contents.

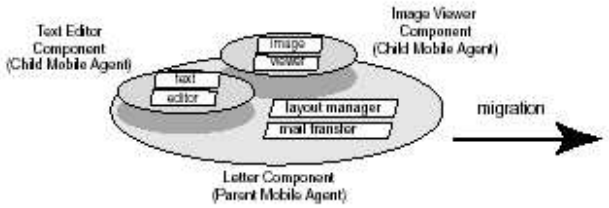

Figure 3: Structure of the Compound Letter Agent

6 Conclusion

There is a strong case for the use of mobile agents in many Internet applications. Moreover, there is a clear evolutionary path that will take us from current technology to widespread use of mobile code and agents within the next few years. Once several technical challenges have been met, and a few pioneering sites install mobile-agent technology, use of mobile agents will expand rapidly. Implementing of the mobile agents is still in progress, in this paper theoretical concepts and some of applications are discussed.

References

[1] Finin, Tim, *KQML Web*, 1998 [online] at http://www.cs.umbc.edu/kqml/

[2] *GeneralMagic, MobileAgents hite Paper*, 1998 [online] http://www.genmagic.com/technology/techwhitepaper.html

[3] Ichiro Satoh *A Formalism for Hierarchical mobile agents*

[4] Harrison et al, *Mobile Agents: Are they a good idea?*, March 28 1995 [online] at http://www.research.ibm.com/massive/mobag.ps

[5] David Kotz and Robert S. Gray, *Mobile agents and the futereg the Internet*, 1999

[6] David Reilly *Mobile Agents Process-migration & in Implication*.

[7] Jonathan Bredin, David Kotz, and Daniela Rus. *Economic markets as a means of open mobile-agent systems*. In Proceedings of the Workshop "Mobile Agents in the Context of Competition and Cooperation (MAC3)" at Autonomous Agents '99, pages 43-49, May 1999.

[8] Amitava Dutta-Roy. *Bringing home the Internet*. IEEE Spectrum, 36(3):32-38, March 1999.

[9] Corey Grice. *When will data change the wireless world*? CNET NEWS.COM, February 10, 1999.

[10] Danny B. Lange and Mitsuru Oshima. *Seven good reasons for mobile agents*. Communications of the ACM, 42(3):88-89, March 1999.

[11] G. Muller, B. Moura, F. Bellard, and C. Consel. Harissa: *A flexible and efficient Java environment mixing bytecode and compiled code*. In Proceedings of the Third Conference on Object-Oriented Technologies and Systems, pages 1-20, 1997.

[12] Giovanni Vigna, editor. *Mobile Agents and Security*, volume 1419 of Lecture Notes in Computer Science. Springer-Verlag, 1998.