A tutorial report for SENG 609.22

Agent Based Software Engineering

Course Instructor: Dr. Behrouz H. Far

Mobile Agents

Submitted by: Eddy Wong

Mobile Agents

As the use of mobile computing devices such as PDAs, laptops, internet-ready cell phones and other similar devices increases, the network environment is becoming more dynamic. Mobile devices connect and disconnect from the network and it is impossible to predict when a network node will be connected or not. Using mobile agents could be the solution to many problems faced when operating in such an environment. Mobile agents are small, customizable programs that can transverse through a network, migrating from one machine to another, completing a task or tasks for the user. The difference between the mobile agent paradigm and other mobile code technologies such as process migration and downloadable applets is that a mobile agent decides when it will move between systems itself. Whereas for process migration, the system decides when to move a process to reduce system load or in the case of applets, the user decides to download the program and executes it on his machine to complete some task.

While many applications can be or is already implemented using a client-server model, the mobile agent paradigm provides a uniform approach to many problems and provides more customizability for the user. With mobile agents, even when the user is disconnected from the network, the agent can continue to work for the user, gathering or monitoring data from different sites and report back to the user when he is connected to the network again. This is extremely useful when a continuous connection is impractical or inconsistent or when bandwidth is an issue, such as in mobile wireless computing using PDAs and cell phones.

One potential advantage of mobile agents is in bandwidth savings. For example, consider a situation where some data from a server needs to be processed and a final result returned. In the traditional client-server model, if there is no single command that can process the data and return the result, then the client must query the server for the data and process it locally. If these intermediate data are not used after obtaining the final result, then the bandwidth usage is potentially wasted, and in a low-speed connection, the computation time will have increased waiting for data. Using a mobile agent to complete the same task, the agent will migrate closer to the server or directly to the server and only send back the final result, saving both bandwidth and time. Also, as stated previously, if the connection is interrupted, the agent could continue completing its task and wait for a reconnection, but with a client-server application, the operation will be stalled or aborted.

Another benefit in using mobile agents is the ability to do load balancing and dynamic deployment. Due to the nature of mobile agents, they could be easily implemented to handle load balancing. As small, self-contained programs that are already programmed to move between various machines “at will”, additional logic could be included to handle load balancing. When the resources of the system that an agent is currently running on are becoming exhausted, the agent can dynamically decide to migrate to another system with a lighter load.

It could be argued that each the benefits of mobile agents could also be implemented and obtained using traditional techniques. However, the advantage of mobile agents lies in the fact that it combines all these strength into a single, uniform approach to solving these problems. In the case of bandwidth savings, a traditional approach might be to use a proxy and for load balancing, it could use a process migration scheme. When both are needed in the same system, combining the two traditional techniques may be difficult, and necessitates the use of mobile agents. There is also another problem, where the clients will need to be customized for different systems that have different features. For mobile agent systems, the agent could be implemented to support the feature that the user needs rather than what the server needs and all that is required is a server that supports the use of mobile agents.

Despite the advantages of mobile agents, there are several challenges and issues that need to be addressed before mobile agent systems will be widely deployed and accepted as a useful computing architecture. Some of the barriers to full-scale mobile agent deployment are discussed in [1]. One of the main issues is in the area of security. There are actually two sides to this problem, both agent security as well as server security. In terms of agent security, the agent must be protected from malicious machines to avoid being corrupted and maintain its integrity. The other security issues deals with protecting the server from malicious agents. With increasing security threats from hackers and the potential for lost revenue from malicious attacks, this is one problem area that must be solved before mobile agents will see wide spread industrial deployment. In an open network system such as the Internet, a system that serves mobile agents could open itself to malicious agents like viruses and worms. And with an agent's ability to spawn sub-components exponentially, it could be use in denial-of-service attacks that renders the system useless.

Another barrier that needs to be overcome is performance. While it is true that mobile agents could complete some tasks faster in the presence of an inconsistent network connection [2], this may not always be the case. Since mobile agents are usually written in interpreted languages such as Java so that it is more portable between platforms, it incurs a penalty in execution time compared with natively compiled code. However, advances in processor speeds and just-in-time compilation are making this less of a problem today. In addition to execution time, mobile agents will also have a migration overhead associated with it. For some applications, the time it takes an agent to pause, transfer itself to another system and resume operation may be significant. Research is currently being done to reduce this overhead. In addition, an agent could decide if migration to another system would be more beneficial than staying in the current system. This technique also shows that a mobile agent is very flexible, becoming mobile only when it is more advantageous.

Standardization is another issue that has to be addressed. It is extremely unlikely that the whole mobile agent community will settle on a single standard and as needs evolve, the standards will change and full backwards compatibility may not always be possible. The agents will need to written so that it is portable between different mobile systems so that

it can access the full range of services provided by different venders using different systems. This, however, will require significant standardization so that the functionally of the agents will not be reduced to the bare minimum.

Besides the technical issues presented above, there are also non-technical issues preventing the wide spread deployment of mobile agents. While technical issues could be overcome in time through research, non-technical issues may not be as easily solved. One of the issues that were mention in [1] was the lack of a "killer application". Without a killer application, most service providers will see little need to convert the current infrastructure to an agent based one. Indeed, most applications for mobile agents today can also be implemented under the current technologies. While mobile agent offers a better solution with more flexibility, it will be hard to convince companies to invest in a new technology that is not significantly superior to the existing proven technology. Most researchers in this field agree that without a killer application and the fact that there may not be one for mobile agents, we will not see a rapid deployment of such systems. However, we will see a trend towards the use of mobile agents system, starting with more specialized needs and it slowly becoming adopted as more and more systems emerges.

This is problem that faces all new technologies. It is the "chicken and the egg" phenomenon. Without agent users, service providers will unlikely be tempted to change their systems and without servers, users will not want to use agents. Therefore, there must be an evolutionary path between client-server systems to full mobile agent systems. There may be proxies that allows the use mobile agents and acts as an interface between agent systems to existing client-server systems. As more agents are deployed and the advantages of such systems become apparent, the servers themselves may open up for direct agent access.

It is likely that mobile agents systems will be used in the local intranet environment first, where it is more controlled compared with the Internet. In such systems, security will be less of an issue as there will be not be any agents from the outside network and the system will be based upon a single standard. This will allow mobile employees to take advantage of an agent system without the system becoming comprised. As more companies embrace agent technology for their own use and as security technology advances, these systems may be opened to allow more interaction with the outside networks, pushing for a global mobile agent system.

Lastly, an important non-technical issue deals with revenue and company image [3]. For a service provider to allow agents to run on their systems, consuming resources and opening them to potential malicious activities, there must be a way for them to recuperate the costs and receive revenue. Today, data providers providing information such as stock quotes recover their costs through advertisements when you visit their web sites. Even now, this revenue model is starting to prove to be inadequate. With the use of agents, you obtain and process this information at the server, most likely bypassing any advertisements the site may have. The actual viewing of their web page will also decrease since the user is using the agent to gather the necessary information. Additionally, by allowing the agents to reside locally on the server, the burden on the

system will also increase. Without a practical way to charge users, these services will simply not be provided. The difficulty is always in the way in which the agents are implemented. Some agents will consume more resources than other, due to poor implementation or additional features and the agents are also designed to be mobile, so that it may choose the best site for any particular task, which may not be always be the same site.

In dealing with company image, the service provider will have little to no control on how their service is presented. It is not necessary true that the service provider is the same as the developer of the agents and their front end. A poorly written agent or front end could be perceived as a poorly provided service, which may or may not the case. The concept of branding the service may not be possible since an agent may be directed to several different services to complete its task.

Currently there are many mobile agent systems being researched and developed. Two of these applications are presented in [2]. One of these is an information-retrieval application. The front end is a GUI on the user's machine that allows the user to enter a text query. When the query is entered, the front end starts the mobile agent on the current machine. Depending on the quality of the network connection, the agent may decide to stay on the local machine. If the connection is poor, the agent migrates to a proxy site. Then it searches the network for the required information. Once the location of the information is found, it can query the information directly or spawn child agents to the information site depending on the size of the information. The child agent gathers and processes the information and sends it back to the parent process. Once all results are obtained, the parent agent merges this information and reports it back to the GUI.

Another application presented was a counter-terrorism application. Each soldier in the team is equipped with a portable computing device with a wireless connection. There will be one or more soldier with equipment that can establish a connection back to their headquarters and act as the gateway for the rest of the team. The soldiers will likely need information from several different sources such as location of other members, description of the suspects, etc. Using a mobile agent system the agent representing a soldier could do a multi-step query, even when the connection is broken. Since during a mission, the soldiers will be in movement and disconnection could frequently occur. Another advantage in using a mobile agent in this instance is that special code could be deployed via agents to the soldiers during an operation since agents does not require any special installation.

An application of mobile agents in wireless network was presented in [4]. This paper shows how mobile agents can use a dynamic routing algorithm to guarantee message transmission in a minimum amount of time in an ad-hoc wireless network with frequent disconnections. Using the agent as a wrapper for the message, the message will move from machine to machine, staying idle if necessary, to send the message to its destination.

In this paper, we have discussed the advantages of mobile agents over conventional technology. The problems and issues preventing the wide spread use of mobile system was also presented. Despite the many hurdles that must be overcome, mobile agents are a promising technology that will allow for a more flexible global network.

References

[1] Milojicic D., "Mobile agent applications", in *IEEE Concurrency*, pp. 80-90, Jul.-Sep. 1999.

[2] Gray R., Kotz D., Cybenko G., and Rus, D, "Mobile agents: Motivations and state-of-the-art systems", at ftp://ftp.cs.dartmouth.edu/TR/TR2000-365.ps.Z.

[3] Kotz D and Gray R, "Mobile Agents and the Future of the Internet", in *ACM Operating Systems Review*, pp. 7-13, Aug. 1999.

[4] Li Q and Rus D., "Sending Messages to Mobile Users in Disconnected Ad-hoc Networks", in *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, pages 44-55, Boston, Aug. 2000.