Empirical Results of the Study for Mobile Agent Technology

Prakash V. Rajguru [1], Mrs. Supriya Yarmal [2] & Mrs. Rajinderkaur Makheja[3]

Abstract: **To provide appropriate agent environment for the studies of developed and researched agent architecture, agent platform were evaluated. First a literature study of existing widely available evaluations had been conducted then it was extended with comparison of platform against proposed criteria related to agent platform. The aim of this research paper is to investigate if one of the possible limits could be the scalability of existing agent environments, For this purpose we have selected JADE agent platform and investigated its performance in a number of test-scenarios. Results of our experiments are presented and discussed. Finally the one platform was selected to implement in Distance Evaluation.**

Keywords: **Itinerary Language, Itinerary Agent, Agent Development Kit, JADE, Grasshopper, Agent Platforms, Scalability, Applications, Agent Communication, Agent, Messaging, licensed under Lesser General Public License (LGPL), Round trip time (RTT)**

1. INTRODUCTION

Mobile agents are autonomous software agents that travel in a computer network to execute and perform tasks on different hosts on behalf of their owners. Autonomous mobile agents bring advantages such as task delegation, network communication, and cost reduction for distributed tasks.

.

1. *Department of Computer Science & IT,*
Adarsh Arts , Commerce & Science College , Hingoli.
prakash_rajgure@yahoo.com

2. *Department of Computer Science & IT,*
Adarsh Arts, Commerce & Science College , Hingoli.
supriya.kandi@gmail.com

3. *Department of Computer Science & IT,*
Adarsh Arts, Commerce & Science College , Hingoli.
silut_st@yahoo.com

2. MOBILE AGENTS AND AGENT PLATFORMS – AN OVERVIEW

Mobile Agents are the software Agents able to roam network freely, to spontaneously relocate them from one device to another.

Software Agents are software components that are [Bellifemine et al., 2003]:

• Autonomous – able to exercise control over their own actions.

• Proactive (or goal-oriented or purposeful) – goal oriented and able to accomplish goals without prompting from a user, and reacting to changes in an environment.

• Social (or socially able or communicative) – able to communicate both with humans and other Agents.

Software Agents operate on Agent platforms. Agent platform is an execution environment for Agents. It supplies the Agents with various functionalities such as Agent intercommunication, Agent autonomy, yellow pages, mobility etc.

Agent platforms are deployed horizontally over multiple hardware devices (such as PCs, PDAs, cellular phones etc) through containers. Each container is an instance of a virtual machine and it forms a virtual Agent network node. On each device *at least* one container may be set up (but there may be much more, like hundreds of them). Containers make Agent platform independent from the underlying operating systems. Mobile Agents are able to migrate from one container to another. Consequently, when containers are deployed on different devices, Mobile Agents can migrate between different devices.

Agent platforms can be imagined as Agent communities where Agents are managed and are given the means to interact. Many Agent communities may coexist at the same time. Depending on the implementation of the platform, Agents may be able to leave one platform and join another.

3. ANALYSIS OF MOBILE AGENT

The situation in the domain of agent technology changes very quickly in the sense that it is quite possible that platforms described one year ago can already is no longer maintained or even unavailable. This is because the technology is relatively novel and mostly in research phase. Academic environments or companies' laboratories for research purposes released the greater number of platforms. Being aware of this fact we formulated the following questions to be answered for each evaluated platform:

Is the platform still maintained?

Is the platform's authors' research group still active?

Is the platform being developed?

Is the platform popular? Is it in broad use?

Is the platform easy accessible?

What is the date of the latest release of platform?

Does a light-weighted release of the platform exist?

How is the platform available?

A. These questions finally helped us in choosing the one platform.

1) Is the platform still maintained?

Agent Development Kit:-Yes

Grasshopper: - No, since November 2003; No longer accessible via URL given on FIPA homepage

JADE: - Yes

2) Is the platform's authors' research group still active?

Agent Development Kit: - Yes. It is commercial company that creates adaptive enterprises through the Distributed Business Process Integration Suite. By using Java technology & services it facilitates complex organizations and business communities to increase their enterprise agility.

Grasshopper: - No. current evidence. The latest accounted activity is dated on October 2003. IKV++ is a company providing business customers with consultancy, information technologies and customized solutions for the provision of communication and information services.

JADE: - Yes. The JADE-Board is a not-for-profit organization with the mission of promoting the evolution and the adoption of JADE by the mobile telecommunications industry as a java-based de-facto standard middleware for agent-based applications in the mobile personal communication sector.

3) Is the platform being developed?

Agent Development Kit:-Yes

Grasshopper: - No recent information evidencing any development activities since last version of platform was released is available.

JADE: - Yes, very actively

4) Is the platform popular? Is it in broad use?

Agent Development Kit: - Tryllian doesn't provide any estimates of the numbers of users, but the platform gained wider popularity after it had become open source

Grasshopper: - No information provided

JADE:-The platform is very popular

5) Is the platform easy accessible?

Agent Development Kit:-Yes, from ADK homepage. For more information about purchasing the ADK 3.0 for commercial use or educational purposes it is necessary to complete and send the Information & Evaluation Request Form and our ADK.

Grasshopper: - No, it is not accessible

JADE: - Yes, from JADE homepage. It is compulsory to fill in the registration form and become a user.

6) What are the version and the date of release of the latest release of platform?

Agent Development Kit:-3.2.0; date not available

Grasshopper: - 2.2.4, date not available

JADE: - 4.1.1 2012

7) Does a light-weighted release of the platform exist?

Agent Development Kit: - No

Grasshopper: - No

JADE: - LEAP

8) How is the platform available?

Agent Development Kit:- Dual-licensed under the LGPL and under a proprietary license

Grasshopper: - Free of charge for non commercial use

JADE: - Open Source; GNU General Public License; It is compulsory to fill in the registration form and become a user

B. Result Analysis

As you can see in point 1-8, restricting the criteria only to the platforms, which were continuously maintained, limited our choice to ADK, Grasshopper and JADE. Then, further on, when we wanted the platform to be also well supported and free, only one framework remained on the field:

JADE. JADE is licensed under Lesser General Public License (LGPL), meaning that users can use both binaries and code of the platform without any restrictions. It is widely used. A light-weighted release of JADE is available the JADE Lightweight Extensible Agent Platform (LEAP). JADE is continuously developed, improved and maintained by the developers from the Telecom Italia Lab (Tilab), where it was originated, and by contributing JADE community members. JADE is also conveniently accessible. The developers and users can download the current version of JADE and additionally the recent snapshots with the latest improvements of the environment [Telecom Italia Lab]. In February, 2007 a comprehensive book on JADE was published: Developing Multi-Agent Systems with JADE [2007], which guides the JADE users through the arcane of programming with JADE.

Since the time we had chosen JADE for the first time we had many occasions to test the platform at work. JADE was the technical platform which served us during our studies on anonymity of software agents, after which we proposed two un traceability protocols and other solutions to support anonymity [2006] and could offer a new add-on for JADE to JADE community of users and developers [2005].

4. PERFORMANCE EVALUATION OF MOBILE AGENT AND CLIENT SERVER TECHNOLOGY

A. Test scenarios

The scenarios centre on pair wise agent-to-agent communication. Different scenarios are realized through change of parameters, which are explained below.

1) Number of hosts **-** The most likely scenario for an agent platform is a distribution of the platform between several hosts. The parameter is either one or two hosts.

2) Number of agent pairs – By increasing the number of agents communicating the general behaviors of the platform MTS as well as scalability are tested. Communication is always considered as a conversation between two agents. So when the numbers of agents are increased it is always done in steps of agent pairs.

3) Message size – Depending on the usage of an agent platform the size of message will vary. Typically the communication is based on interactions with relatively small messages and therefore the tests with increasing number of agents are using 2 KB messages. Cases with messages of larger size are possible and this has also been tested as single agent-pair conversations with message sizes between 0 and 100 KB.

4) Message encryption **-** This is one way to test performance-security trade-offs. Platform configuration, were considered in initial experiments and suitable settings were established. For the experiments presented here, those parameters were fixed for each platform.

B. Test bed

The tests were done on a single computer and on a Local Area Network (LAN) consisting of two computers connected with a 100 Mbps cable. The two computers share the same fundamental set-up according to Table 1. A separate network was used to avoid influence of other hosts. Computers used in the experiments.

Operating System: Microsoft Windows XP
Java VM : Java SDK 1.4.2 (1.3.1 for JADE)
CPU : Dual Core
Memory : IGB (DDR)
Network card : Ethernet 10/100

As all three-agent platforms are implemented in Java so were also the agents used for the experiments. The latest Java SDK version 1.4.2 with accompanied runtime environment was used but also version 1.3.1, as one of the platforms, did not support any later versions. The Java Virtual Machine and platforms were restarted between each experiment to avoid influence of previous experiments.

C. Measurement

The clock used is Java's System. Current Time Millis () method and the unit is milliseconds. The resolution of the clock (10 ms) is considered good enough since time is measured for 1000 and 10000 repetitions rather than individual messages. For each experiment the sender agent sends a message to the receiver agent. The receiver agent obtains the payload, and sends an equivalent message back. This is repeated 1000 times for single agent pairs and the mean RTT is used as result. To obtain good confidence in the results, each experiment was repeated ten times. The average is presented here. Single pair experiment is straightforward to implement and test. With multiple agent pairs there is a short period in the beginning (end) of each experiment when not all agent pairs have started (finished) exchanging messages. By increasing the number of exchanged messages from 1000 to 10 000 for all experiments with multiple agent pairs the influence on the results becomes insignificant.

5. EXPERIMENTS

We implemented client/server and mobile-agent versions of a simple information-retrieval application and used query completion times as the performance metric for comparing them. The application is a simplified version of the examination system tasks that the student in the field performs against the Examination databases. Specifically, the application allows a user to retrieve a set of Question Paper from Distributor server. Our intention, however, was not to measure the performance of a domain-specific retrieval application, but instead to measure the performance of a simplified application that (1) allowed straightforward control over parameters such as number of student, Question paper, submitting answer, and so on, and (2) was representative of many domain-specific applications. Our simplified application fills both roles quite well, and in particular, is an effective stand-in for any application that uses a service's built-in operations to obtain a set of candidate documents, but then performs a single pass over the candidate documents, either to reduce the number of candidates or the size of each candidate. For the Distance Evaluation, for example, the client application used built-in service operations to perform standard keyword and phrase searches, but then automatically constructed summaries of the candidate documents.

Second, the agents in the mobile-agent version of the application make only a single migration from client to server machine. The decision of whether to make a single initial migration, however, is the first and most important decision in any mobile-agent application. Can the client code efficiently interact with the service from its home machine, or should it send some part of itself to a more attractive network location? Analyzing the scalability of the single-migration case is an essential step toward an understanding of mobile-agent performance.

Finally, as we will see, the performance curves turn out to be straightforward, and can be described easily with simple mathematical equations. In fact, as long as the agent system itself is not overloaded, the query completion time's scale linearly according to the amount of data transferred over the network, which in turn is proportional to the number of client machines making simultaneous queries.

It is important to verify that a real mobile agent system does meet our scalability expectations, so that application designers can confidently analyze situations for which experiments are difficult or impossible. The agent version of our application uses a parent agent to launch from one to twenty identical agents, each to a separate workstation. Each agent acts as a client in an examination system task in which it jumps to a central server, does a single initial query against a document collection to get a list of candidate documents, then searches the text of the candidate documents for a particular substring, and then returns to its home machine with the matching documents. The time it takes for the processing and jumps of each agent is calculated and reported to the parent agent where they are averaged and summarized as a performance measurement. By using one workstation per client agent, we ensure that we are measuring the scalability of the server and network, rather than of the client machine. The results of our performance experiments would provide important information, however, to actual applications that need to decide whether to use a proxy site and whether to spawn child agents. In the client/server version of the application, the client opens a TCP/IP connection over which it sends queries, using a simple protocol, to a multithreaded. The server performs this initial query to get a list of candidate documents and then sends all the resulting documents back to the client. The client then does a substring search on the documents to select the final results. The clients and servers keep track of the connection duration and substring search times as a performance measure. We ran our experiments on twenty computers, the one to twenty client machines were connected to the one document-server machine through a 100 Mbps switch and hub, a hardware bandwidth manager set to workstation, thus the client machines share the given bandwidth. The authentication features of Agents were turned off, so both the agents and the client/server messages were unencrypted and unsigned.

The text of each document is exactly 4 kilobytes in size, but each retrieved document is 4.2 kilobytes, since the retrieved document includes a URL, a unique integer identifier, and other meta-information. In the agent case, each retrieved document actually takes up 4.5 kilobytes, an additional 0.3 kilobytes, due to the details of how Agents packages up agents for transmission. The Java class file that contains the mobile Java agent is 3.2 kilobytes. 3.2 kilobytes of code may, a simple Java agent. This agent migrates back and forth between a client and a server machine, performing an information retrieval task on the server machine and returning the results to the client machine.

.

6. DISCUSSION & CONCLUSIONS

In this Research, we described the Mobile Agents system, which supports multiple languages, communication, security, resource control, and strong migration, and quantified its performance in distributed information retrieval applications. Our experience with the use of Mobile Agents in a large application that supports Examination System in the field shows that the mobile-agent paradigm works. Mobile agents are a useful approach for many distributed applications. A mobile agent does not require a permanent connection to the computer from which it was launched. The agent does not care if the computer that launched it becomes disconnected because it can pursue its search operations independently. Moreover, agents are an efficient paradigm for information processing and transfer over wireless networks, which typically have low bandwidth and high latency. By migrating to the location of an electronic resource, an agent can access the resource locally and eliminate costly data transfers over congested networks. This reduces network traffic and improves data delivery, because it is often faster and cheaper to send a small agent to a data source than to send all the intermediate data to the requesting site. In short, we see six key benefits of mobile agents: (1) conservation of bandwidth, (2) reduction in latency, (3) reduction in total completion time, (4) support for disconnected operation in mobile computing, (5) load balancing, and (6) dynamic code deployment. This Research focused on bandwidth conservation and reduction in completion times. Thus, the true strength of mobile agents is that they provide a general-purpose framework in which a range of distributed applications can be implemented easily and efficiently.

Mobile Agents is an excellent prototyping tool for distributed applications, but several challenges must be overcome to fully realize the six mobile-agent benefits. Specifically, the protocol overhead of the Mobile Agent paradigm is large, which has significant performance consequences. Due to the migration and interpretive overhead in our system, an agent will outperform a traditional client/server solution (in terms of total completion time) only if a query retrieves a sufficiently large number of documents and the agent is able to filter most of them on the server side of the connection. This observation is particularly true if the client and server machines are connected with a medium-speed or faster network (such as a 10 Mbps Ethernet). From a scalability viewpoint, even though the mobile agent approach does better than the client/server approach when there are more clients in the system, it is easy to see that scalability becomes an issue for mobile agents as well when the maximum channel bandwidth is reached. In addition, mobile agents are more likely to overload the CPU on the server machine, and we must understand whether bandwidth or the server CPU will be the limiting factor in any given application. Thus, while many difficult problems have been solved for mobile agents, many challenging research problems remain. We believe that the key research areas are scalability, security, and program verification, and our projects focus on scalability, the first of these three issues. Our future goals for Mobile Agents are to

(1) Support agent migration that is only a small factor slower than an RPC call that transmits an equivalent amount of data,

(2) Execute agents nearly as quickly as if they were compiled into native code,

(3) Integrate a high-performance encryption library, and, most importantly,

(4) Expand on the scalability experiments and develop general models for mobile-agent performance,

Particularly for Distance Evaluation applications, Such models will help a designer decide whether to use mobile agents at all, and will help mobile agents dynamically decide what to do given current network and machine conditions.

. 7. REFERENCES

[1] Colin G. Harrison David M. Chess Aaron Kershenbaum, "Mobile Agents: Are they a good idea?",IBM Research Division T. J. Watson Research Center Yorktown Heights

[2] Danny B. Lange and Mitsuru Oshima, "Seven Good Reasons for Mobile Agents", Communications of ACM, Vol. 42, No. 3

[3] David Camacho, Ricardo Aler, 2002, "Performance Evaluation Of Zeus, Jade,And Skeletonagent Frameworks" Computer Science Department, Universidad Carlos.

[4] Foner, L. (1993), "What is an Agent, Anyway? A Sociological Case Study", Agents Memo 93-01, MIT Media Lab, Cambridge, MA.

[5] Genco, 2008, "Mobile agents: principles of operation and applications", 271 pages

[6] Krzyszt of Chmiela, Dominik Tomiaka, at all, "Testing the Efficiency of JADE Agent Platform"

[7] Kalle Burbeck, Daniel Garpe, Simin Nadjm-Tehrani, "Scale-up and Performance Studies of Three Agent Platforms"

[8] Maes, P, "Agents that Reduce Work and Information Overload", Comm. ACM 37(7): 31

[9] Nupur Giri, Shrikant Bodhe, May 2009, "Performance Evaluation of Multi Agent Based Call Admission in Cellular Networks" International Journal of Recent Trends in Engineering, Vol 1, No. 1

[10] Rocco Aversa, Beniamino Di Martino, Massimiliano Rak and Salvatore Venticinque, "A Framework for Mobile Agent Platform performance Evaluation "

[11] R. Aust and R. Isaacson. 2007, "Designing and Evaluating User Interfaces for eLearning." In: G. Richards (Ed.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, pp 1195-1202.

[12] Roy Clariana and Patricia Wallace, 2002 "Paper Based versus Computer based assessment: Key factors associated with the test mode effect",British Journal of educational technology Vol. 33 No.5 p. 596-602

[13] Rahul Jha and Sridhar Iyer, "Performance Evaluation of Mobile Agents for E-Commerce Applications" Hewlett Packard Laboratories

[14] S. Franklin and A. Graesser, "Is it an agent, or just a program?" A Taxonomy for Autonomous Agents" Institute for Intelligent Systems University of Memphis Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag

[15] Syed Adnan John Datuin, June 13, 2000, "A Survey of Mobile Agent Systems" Final Project

[16] Yuan Ze,Diana, 9th April 2003 "Agent Platform Evaluation"

[17] Xining Li Systemics "Comparison of Communication Models For Mobile Agents", Cybernetics And Informatics Volume 1 - Number 2

[18] Jingqiao Zhang, September 1, 2009, "JADE: Adaptive Differential Evolution with Optional External Archive"