Comparative Advantage of Mobile Agent Application in Procuring Software Products on the Internet

Michael K. Adu, Boniface K. Alese, Olumide S. Ogunnusi

Abstract—This paper brings to fore the inherent advantages in application of mobile agents to procure software products rather than downloading software content on the Internet. It proposes a system whereby the products come on compact disk with mobile agent as deliverable. The client/user purchases a software product, but must connect to the remote server of the software developer before installation. The user provides an activation code that activates mobile agent which is part of the software product on compact disk. The validity of the activation code is checked on connection at the developer's end to ascertain authenticity and prevent piracy. The system is implemented by downloading two different software products as compare with installing same products on compact disk with mobile agent's application. Downloading software contents from developer's database as in the traditional method requires a continuously open connection between the client and the developer's end, a fixed network is not economically or technically feasible. Mobile agent after being dispatched into the network becomes independent of the creating process and can operate asynchronously and autonomously. It can reconnect later after completing its task and return for result delivery. Response Time and Network Load are very minimal with application of Mobile agent.

Keywords—Activation code, internet, mobile agent, software developer, software products.

I. INTRODUCTION

MOBILE agents open several new possibilities for conducting business in a network and especially the internet environment. It can therefore serve as a tool for procuring software products on the Internet. Permanent connection to the Internet is not necessary as a result of the mobility and autonomy attributes of mobile agents. Software agents brought about the concept of mobile agent [1]. The concept of software agents is based on objects as recognized in the object-oriented environment and they can adopt several different forms such as stationary agents, intelligent agent and mobile agents. According to [2], the concept of having software entities that can carry out some set of operations for the user or for another program is described as intelligent agents. It carries out its operation with some degree of autonomy or independence. Mobile agents however, are computer programs, which are autonomous, proactive and reactive, and have ability to learn. They move from one node to another node and interact with each other, sharing information to better carry out their tasks. Mobile agents spread intelligence across network [3]. Major actions on mobile agents which include creating, deployment and termination of activities are made possible by the mobility attribute and without any effect on the network configuration [4].

An agent is generally as mobile when its execution can be interrupted (usually briefly), before it migrates to a new host and then continues its activities at the new runtime environment [5].

Mobile agent is an autonomous program that moves between networks to take advantage of the service supplied by stationary agents [6]. Mobile agents carry the burden of computation or activities close to the resources they need to access rather than transporting same over the Internet. This creates the benefits of reduction in both the network communication bandwidth and latency. Therefore, most of the challenges of distributed system today can be handled by using mobile agents to overcome the limitations to meet requirements. Mobile agents are deployed for various purposes.

In this article, the existing application of mobile agents in electronic commerce and other areas is summarized in Section II to provide insights to its application in other critical areas especially with regard to procurement of software products on the internet. In Section III, comparative test and evaluation of mobile agents' application in procurement of software products was carried out. A conclusion is drawn in Section IV.

II. APPLICATION OF MOBILE AGENTS

The mobile agent paradigm has been very useful in many application areas, some of the most salient include:

A. Electronic Commerce

One area that has benefitted immensely from the application of mobile agent is electronic commerce (e-commerce). Certainly, *electronic commerce* is one of the most attractive areas to researchers and industries. Given a widely deployed mobile agent infrastructure for electronic commerce, mobile agents may contribute in three ways:

Michael K. Adu is a Principal Lecturer in Computer Science Department, the Federal Polytechnic, Ado Ekiti, Nigeria (phone: +2348066714060; e-mail: memokadu@yahoo.co.uk).

Boniface K. Alese is a Professor in Computer Science, Department and Dean, Students' Affairs, the Federal university of Technology, Akure, Nigeria (phone: +2348034540465; e-mail: bkalese@futa.edu.ng).

Olumide S. Ogunnusi is a Chief lecturer in Computer Science Department, the Federal Polytechnic, Ado Ekiti, Nigeria (phone: +2348030695802; e-mail: olu_simmy@yahoo.com).

1. Optimizing Decisions

In [7] and [8], they opined that electronic markets show increasing market transparency resulting in superior resource allocation. However, transparency may be a difficult task in such an interactive electronic market dealings, but with application of mobile agents, optimized decisions for allocation of goods can be achieved via economical and technical means for high speed and effective information searching.

2. Providing Mobility, Flexibility and Autonomy

Mobile agent can migrate easily from one node to the other in a distributed environment and has high attribute for flexibility by cloning itself and can operate autonomously by acting independently. These are of great benefits to the customers as well as merchants [9].

3. Increase Market Efficiency for All Parties

Mobile agents are very efficient and effective in saving time. In most cases this is achieved by eliminating other agents that are not necessary for a particular transaction.

B. Other Major Areas of Application

Several other applications that benefit from the use of mobile agent systems include information searching, filtering and retrieving, to electronic commerce on the Web where they, as personal assistants for their owners as well as piracy prevention of software products [10]. Education, workflow management, and information retrieval management, among others, are other uses of mobile agents [11].

III. COMPARATIVE TEST OF MOBILE AGENT'S APPLICATION

For the purpose of this research work, which is to bring to fore the better performance of mobile agent application as against client-server or Remote Procedure Call (RPC) by downloading software contents on the internet, our measurements were obtained by the use of eight personal computers placed at different locations across western Nigeria, each provided with a copy of VLC and Digsby software with mobile agent as deliverable. They were provided with activation-codes generated that corresponded with the software for the purpose of this work only and were maintained at the sever end. They also installed same software products by downloading the software content on their computers to demonstrate Client–Server approach without mobile agent application.

A. Test Environment

The eight computer systems were placed at the Federal University of Technology FUTA (Akure), Crown Polytechnic, Odo (Ado–Ekiti), Continuing Education Centre, Federal Polytechnic, Ilawe Road, (Ado-Ekiti), Federal University of Technology, Students' Affairs Office, FUTA (Akure), Oba Akran Avenue, Ikeja (Lagos), Obafemi Awolowo University (Ile-Ife), Memolinks Computer College, Ilere, Oke-Odu, Ijare Road (Akure), Plot II, Block B Arofi Estate, OritaObele, Ijare Road (Akure). Information about the computer systems used as collected and maintained at developer's server is presented in Table I.

TABLE I
COMPUTER SYSTEMS USED AS COLLECTED AND MAINTAINED AT DEVELOPER'S SERVER

PC NAME	HDD FILE SYSTEM	C-DRIVE SIZE	PROCESSOR NAME	PROCESSOR ID
BAYO ADETUNMBI	NTFS	622349774848	Intel® Core™ 15-2430M CPU 2.40GHz	BFEBFBFF000206A7
CURIOSITYMASTER	NTFS	115238498304	Intel ® Atom (™ CPU N450 @ 1.66GHz	BFE9FBFF000106CA
CHUMYLEE-PC	NTFS	119926681600	Intel ® Core ™2 Duo CPU T7100 @	BFEBFBFF000006FD
JERAND-PC	NTFS	484272263168	Intel® Core ™13 CPU M 370 @ 2.40GHz	BFEBFBFF00020655
NACOSSCECFEDPOL	NTFS	367001595904	Intel® Pentium(R) CPU G2020@2.50Hz	BFEBFBFF000306A9
AKINWALE-PC	NTFS	138962530304	Intel ® Celeron ® CPU 900 @ 2.20GHz	AFEBFBFF0001067A
USER-PC	NTFS	78080212992	Intel ® Celeron ® CPU 900 @ 2.20GHz	AFEBFBFF0001067A
MEMOLOINKS	NTFS	500000878592	Intel® Pentium(R) CPU G2020@2.50Hz	BFEBFBFF0000306A9

B. Evaluation Criteria

In this comparative model, the effectiveness of any of the two approaches over the other would depend on how best it could manage network resources in term of minimal network load and response time. The two quantifiable measurable quantities were chosen as evaluating metrics; that is, response time and the network load. These two metrics were employed to compare the performance of the RPC approach versus the mobile agent-based paradigm. This would enable us to decide mathematically which paradigm produces less network traffic in order to provide most rapid response in line with the objectives of this work.

C. The Performance Model

The mobile agent is a small software entity that can deliver even when the network goes down unlike when the whole

software content is downloaded as in [12], an RPC approach, which makes it almost an impossible task when the network goes down. Also as illustrated in Fig. 2, mobile agent application enables the data to be processed in its locality, more importantly when large volume of data is involved, without necessarily having to transfer it over the Internet. This reduces the flow of raw data in the network. [13]. In this research work, the mobile agent only confirms the authenticity of the software product and installation takes place at the client/user's end unlike the case of client-server/RPC approach, the software content is at the developer's server, downloaded over the Internet.

The network load in bytes and response time in milliseconds for this work was arrived at mathematically and compared with software content downloading through RPC. An analytical model that describes the network load and the

response time in order to compare the performance of both the mobile agent application in procuring software products and
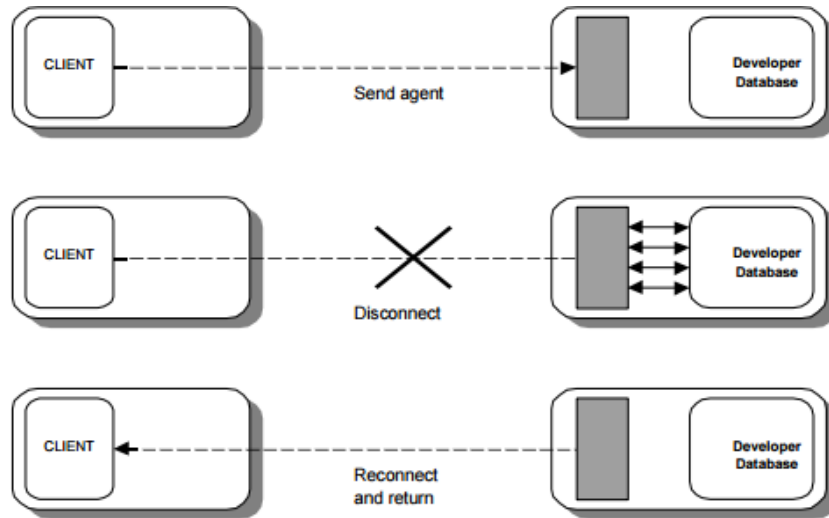

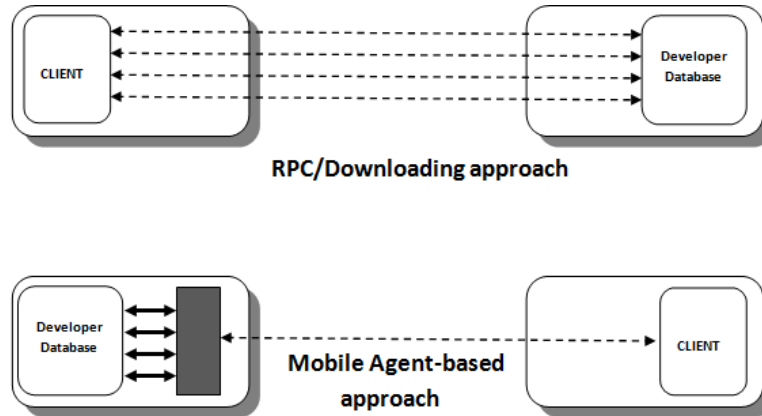Fig. 1 Mobile agents allow disconnected operations

Fig. 2 Mobile Agents Reduce Network Load

D.Network Load

The network load refers to the total amount of data in bytes moved across the Internet.

1. Network Load for Mobile Agent Application

When agent migrates to the server
LOAD = code (B_c bytes) + State information (B_s bytes) + Request object i.e. Activation code and PC identity information

$$(B_{req} \ bytes) = B_c + B_s + B_{req} \tag{1}$$

At the server, agent only communicate, no load. Agent filters and compresses data at the server by ratio σ , therefore only (1- σ) B_{res} is carried back home as response.

Agent does not carry request back home, hence Breq is no longer part of the load back.

Since the code B_c resides at the user end, it is also no longer

client-server approach or RPC is presented. Common application areas were considered for both scenarios.

part of the load back, only the state information B_s is carried back. Therefore LOAD at the server back home is

$$B_s + (1 - \sigma)B_{res} \tag{2}$$

Total Network Load
$$B_{MA} = (1) + (2)$$
$$= B_c + 2B_s + B_{req} + (1 - \sigma)B_{res} \tag{3}$$

2. Network Load for RPC/Client-Server by Downloading
User sends a request of B_{req} bytes (Activation code)
The response is neither filtered nor compressed. When the code is valid and content of software is downloaded from server to user Pc, then total network load,

$$B_{DL} = B_{req} + B_{res} + S_c \tag{4}$$

where S_c is the software content in bytes.

TABLE II
NETWORK LOAD FOR MOBILE AGENT APPLICATION VERSUS RPC/CLIENT-SERVER FOR VLC SOFTWARE

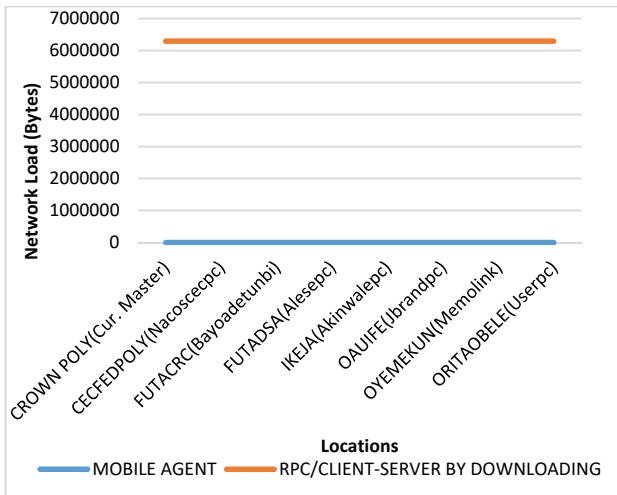| Location/PC Name	Mobile agent	Downloading
CROWN POLY(Cur. Master)	70	6291456
CECFEDPOLY(Nacoscecpc)	70	6291456
FUTACRC(Bayoadetunbi)	70	6291456
FUTADSA(Alesepc)	70	6291456
IKEJA(Akinwalepc)	70	6291456
OAUIFE(Jbrandpc)	70	6291456
OYEMEKUN(Memolink)	70	6291456
ORITAOBELE(Userpc)	70	6291456

Fig. 3 Location vs. Network Load (VLC)

TABLE III
NETWORK LOAD FOR MOBILE AGENT APPLICATION VERSUS RPC/CLIENT-SERVER FOR DIGSBY SOFTWARE

Location/PC Name	Mobile Agent	Downloading
CROWN POLY(Cur. Master)	70	10066329.6
CECFEDPOLY(Nacoscecpc)	70	10066329.6
FUTACRC(Bayoadetunbi)	70	10066329.6
FUTADSA(Alesepc)	70	10066329.6
IKEJA(Akinwalepc)	70	10066329.6
OAUIFE(Jbrandpc)	70	10066329.6
OYEMEKUN(Memolink)	70	10066329.6
ORITAOBELE(Userpc)	70	10066329.6

TABLE IV
EXECUTION TIME IN MILLISECONDS (MOBILE AGENT APPLICATION VERSUS DOWNLOADING BY ACTIVATION CODE BY REUBEN, 2010, IMPLEMENTED WITH VLC MULTIMEDIA SOFTWARE)

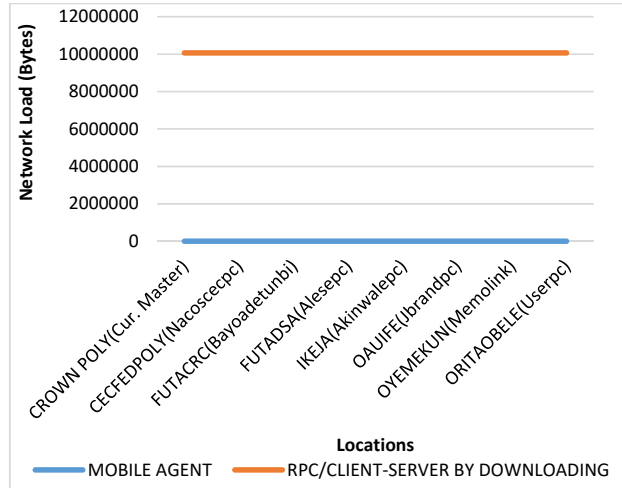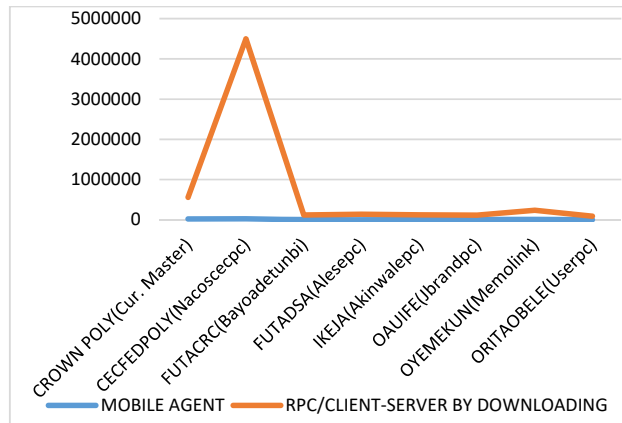| Location/PC Name	Mobile Agent	Downloading
CROWN POLY(Cur. Master)	20140	560100
CECFEDPOLY(Nacoscecpc)	23300	4500000
FUTACRC(Bayoadetunbi)	3229	120105
FUTADSA(Alesepc)	2537	140186
IKEJA(Akinwalepc)	4179	123540
OAUIFE(Jbrandpc)	5130	118250
OYEMEKUN(Memolink)	9285	240000

Fig. 4 Location vs. Network load (DIGSBY)

Fig. 5 Location vs. Response Time (VLC)

TABLE V
EXECUTION TIME IN MILLISECONDS (MOBILE AGENT APPLICATION VERSUS DOWNLOADING, IMPLEMENTED WITH DIGSBY MULTIMEDIA SOFTWARE

Location/PC Name	Mobile Agent	Downloading
CROWNPOLY(Cur. Master)	20250	561100
CECFEDPOLY(Nacoscecpc)	23650	4515500
FUTACRC(Bayoadetunbi)	3344	120605
FUTADSA(Alesepc)	2655	140986
IKEJA(Akinwalepc)	4329	124740
OAUIFE(Jbrandpc)	5286	119050
OYEMEKUN(Memolink)	9495	247000
ORITAOBELE(Userpc)	3202	105000
ORITAOBELE(Userpc)	3122	90000

E. Response Time

This is the total response time in seconds.

1. Mobile Agent Application

Time to marshal one byte is T_m, time to process the request at the server is T_p, and constant overhead scheduling time is T_s. Response time in seconds is:

$$T_{MA} = B_{MA}(T_m) + T_p + T_s \qquad (5)$$

2. RPC/Client-Server by Downloading

Response time in seconds is:

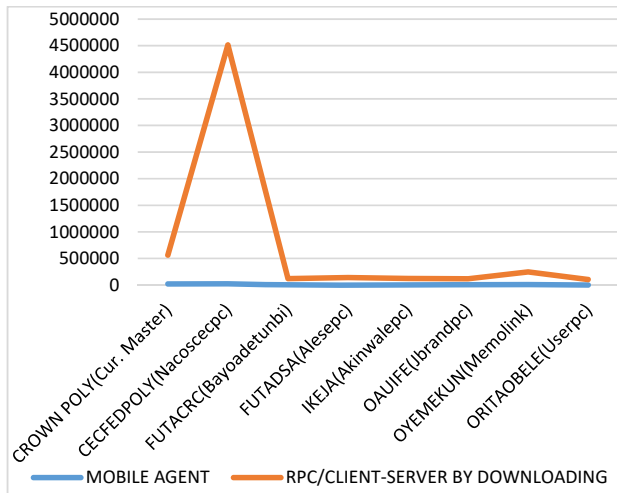$$T_{DL} = B_{DL}(T_m) + T_p + T_s \qquad (6)$$

Fig. 6 Location vs. Response Time (DIGSBY)

IV. Conclusion

This work emphasises the fact that efficiency of the network in a distributed environment can be enhanced by application of mobile agent. It is pertinent to observe that individuals and organisations nowadays have their operations carried out with the use of network facilities and the internet technology for optimal results. Collective efforts to minimize wastefulness of resources to ensure network availability and sustainability are highly germane. It is obvious that this will soon place all users/buyers of software products to purchase and confirm authenticity against piracy on the Internet. The issue of security impediments are being looked into by researchers who are making relentless efforts to overcome the challenge, this will not only bring about optimal result, but the interesting properties inherent in mobile agent application would be put to use. This work has apparently brought into fore the need for further and urgent research work in application of mobile agent in procuring software products with the involvement of software developers.

References

[1] Giovanni Vignas, "Mobile Code Technologies. Paradigms, and Applications," Ph.D thesis, Polilecnico Milano, 1995.
[2] Robert S Gray, David Kotz, and Ronald A Peterson, "Mobile-Agent versus Client/Server Performance Scalability in an Information Retrieval Task," Proceedings of the Fifth IEEE International Conference on Mobile Agents, Atlanta, Georgia, 2001, pp. 229-24.
[3] Shug-Pyng, Chern-Tang, L and Shianyow, W, "Optimal Assignment of Mobile Agents for Software Authorization and Protection", National Science Council,Taiwan, 2004.
[4] Djekic, P., and Loebbecke, C. "*Preventing application software piracy: An empirical investigation of Technical Copy protection", The Journal of Strategic Information Systems, 2007, 16(2), 173-186.*
[5] David, K and Robert S, "Mobile Agents and the Future of Internet" Seatle, Washington, U.S.A, 1999.
[6] Antonio Carzamga, Gian Pietro Picco, and Giovanni Vigna, "Designing Distributed Applications with A Mobile Code Paradigm," Proceedings of the 19th International Conference on Software Engineering, 1CSI397, Seattle USA, 1997, pp 22' 32.
[7] Malone, T. W., Jates, J., and Benjamin. R. I, "The Logic of Electronic Markets", 1989, Harvard Business Review 67, 3: 166-172.
[8] Bakos, J, "A Strategic Analysis of Electronic Merket places", 1991, MIS Quarterly 11,4: 295-310.
[9] Bernichi, M and Mourchi, F, " Software Management Based on Mobile Agents", Proceedings in the International conference on Instrumentation, Communication and Information Technology(ICIC), August 3rd -5th, Indonesia, 2005.
[10] Tripathi, A. R., Karnik, N. M., Vora, M. K, Ahmed, T. and Singh, R.D, "Mobile Agent Programming In Ajanta", In Proceedings of the 19th International Conference on Distributed Computing Systems (ICDCS '99), Austin, Texas, 1999, pp. 190-197.
[11] Grimley, M. J. and Monroe, B. D., "Protecting the Integrity of Agent: An Exploration in Letting Agents loose in an Unpredictable World", (Online) ACM crossroad, ACM. Available at: http://www.acm.org/crossroads/xrds5-4/integrity.html>, 1999.Accessed on 04/01/15.
[12] Reuben B, "Activation Code System and Method for Preventing Software Piracy," United States Patent Application Publication, Bahar. 2010.
[13] Adu, M. K., Alese, B. K., and Adewale, O. S. "Design of Software User Identity Module (SUIM) for Preventing Software Piracy", World Congress on Engineering, International Association of Engineers (IAENG), London, U.K, 2014.