

An Efficient E-Commerce Application Development Using Mobile Agents

Mahesh Kumar KondReddy

*Dept. of Computer Science & Engineering,
SV University , Tirupati-517502.
Andhra Pradesh,India*

Sujeeth.T

*Dept. of Computer Science & Engineering,
SV University , Tirupati-517502.
Andhra Pradesh,India*

Abstract- Mobile Agent (MA) paradigm has received a great deal of attention in distributed computing systems. Mobile agents are defined as objects that have behavior, state, and location. A mobile agent is a program that can autonomously migrate between various nodes of a network and perform computations on behalf of a user e-commerce applications involve interaction among entities running in different machines. Mobile agents can be used in e-commerce applications which access the data locally by moving code to the machine and executing there .We developed a e-commerce application for online book store using client-server model and mobile agents. Experimental results show that mobile agents are efficient taking less turnaroundtime than client-server model .

keywords: Mobile Agents, e-commerce, agent

I. INTRODUCTION

Mobile Agent (MA) systems have for some time been seen as a promising paradigm for the design and implementation of distributed applications. A mobile agent is a program that can autonomously migrate between various nodes of a network and perform computations on behalf of a user. Some of the benefits provided by MAs for creating distributed applications include reduction in network load, overcoming network latency, faster interaction and disconnected operations[1].A mobile agent is a program, which represents a user in a computer network, and is capable of migrating autonomously from node to node, to perform some computation on behalf of the user.

Mobile agents are defined as objects that have behavior, state, and location [2]. Its tasks are determined by the agent application, and can range from online shopping to real-time device control to distributed scientific computing. Applications can inject mobile agents into a network, allowing them to roam the network either on a predetermined path, or one that the agents themselves determine based on dynamically gathered information. Mobile agents represent a class of agents whose main functions are their transmission capabilities between nodes on the same network or different networks, in addition to the inherited capabilities of stationary agents. They represent the basic design agent information management.

Mobile agents represent a direct extension of the client server approach. In the client-server paradigm [2], the communication entities have a role well defined and fixed. The server offers a set of services and the client uses these services. This implies a dependency of client on services provided by the server. The mechanism of transmission that occurs between a client and a server is performed by a message. With this method it is necessary to the programmer to program the server address and to provide synchronization between the two entities. The system RPC (Remote Procedure Call) was developed by Sun Microsystems Inc. [3] to simplify the operating of client server by simplifying the programming. The services introduced allow the customer to feel like making a local application. These services are represented by stubs.

The location of the server, triggering the transport service and results are handled in a transparent manner to the client. However, a fundamental problem exists with the approach client/server with regard to the management of distributed information. If the server does not provide the exact service that the customer needs, for example the server only provides low levels, then the client must continually poll the server. This can lead to increased waiting

times. In summary the RPC approach: Uses bandwidth network for each message; Requires maintaining communication with the network with a specific service for the duration of communication. Other client/server architecture are CORBA (Common Object Broker Architecture) [5] who wants to make the paradigm client/server accessible by adopting the object-oriented principles (reuse of object, inheritance and the encapsulation, ...) and the Distributed Computing Environment RPC (Open Software Foundation, 1992), which provides equipment and security and authentication interface using threads instead of sockets to achieve a higher level of abstraction.

Some advantages of mobile agents are : A mobile agent can continue to run after leaving a node, even if they lose connection with the node where they were created or from originating node. Agent reduces traffic in the network by doing the necessary processing in the remote node and transferring only the results to host node. Agents are ideally suited to build parallel distributed applications where agent can move on to other machines when necessary and can delegate tasks to other mobile agents in order to achieve real parallel applications. Mobile agents are fault tolerant to network failures , moves from one node to another without any intervention and meets predefined goals

II. EXISTING WORK

In traditional e-commerce applications ,a client-server model is used to communicate with machines in a distributed system. for example in an online application for book store with ten shops a separate request is made to each node and the until data is received connection is to be active. In mobile agent based architecture a mobile agent is created based on user preferences selected and moves across each seller machine , executes and after completing all the nodes it returns to the buyer's system with the results collected at each node . Mobile agent paradigm provides a cleaner design for several real life application as compared to the traditional client server model. The mapping of implementation components to real life objects is direct while using mobile agent technology[9] .

With the introduction of Java to the Internet world, many mobile agent projects have made use of this operating system independent language. Another benefit to using Java is that each of these systems can make use of the standards that are inherent in Java such as the Java virtual machine and object serialization mechanism . Some of these systems are listed below :

- *Aglets*, IBM's mobile agent system. The word Aglet is formed through the combination the words agent and applet, as the intention of this system is to bring mobility to Java applets [6].
- *Odyssey*, from General Magic Inc. was the first mobile agent system. It was reworked using Java and now provides a set of Java classes that developers can make use of to create their own mobile agent applications[7].
- *Concordia*, Mitsubishi's agent system which provides developers with a framework for the development and the management of mobile agent applications[8]. These applications can be extended to any system supporting Java.
- *Voyager*, an agent based system that supports both traditional and agent-based distributed computing techniques created by Object Space. Voyager supports object request brokering so developers can create distributed application using both traditional messaging, such as RMI or CORBA, as well as agent-enhanced techniques [1].

III. PROPOSED SYSTEM

The proposed system implements a complete business-to-customer e-commerce based application using mobile agent technology. The proposed model address the design issues with mobile agent in e-commerce as well as the software engineering aspect of mobile agent technology. In our prototype model we have created an e-market place for buying and selling of goods using both the traditional client server paradigm and the mobile agent paradigm. e-shops were hosted which handle large database of products . user searches for a book with his preferences specified as attributes across shops of interest. Products are represented in XML files and user preferences are also specified as an XML file which searched in the product catalog. The proposed system is implemented by using Voyager object-request-broker framework for mobile agents. In Voyager an agent is a special kind of object that can move independently, can continue to execute as it moves, and otherwise behaves exactly like any other object. Voyager enables objects and other agents to send standard Java messages to an agent even as the agent is moving.

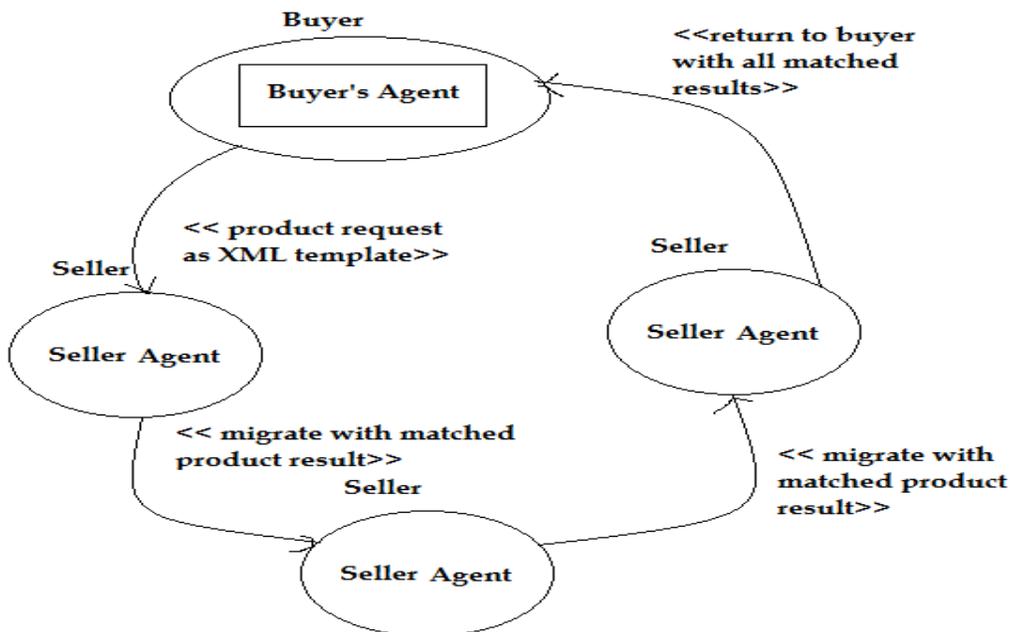


Fig: Buyer 's purchase scenario of selected products

In addition, Voyager allows to remote enable any Java class, even a third-party library class, without modifying the class source in anyway. Voyager also includes a rich set of services for transparent distributed persistence, scalable group communication, and basic directory services. Buyer launches a mobile agent based specifying a list of shops to be visited , preferences and the product evaluation logic. The buyer's agent on behalf of buyer visits each shop in the specified order executes there to find the user specified data. Buyer's agent invokes the services provided by one or more Data Access Objects to search for the specified objects. Using the constraints imposed by the buyer , the search result is filtered .The buyer agent then moves to to shop along with this information . After completing the search operation in all the user specified shops buyer's MA returns back to the buyer's site and then displays the information to the user and the best deal to the user. The user can make a purchase based on the data returned by the mobile agent.



fig: Plot of turnaround time at each hop(e-shop)

Mobile agents executes the evaluation logic based on the user preferences and returns only those products thus reducing network load. The time duration between after initiating a request and the response received , turnaroundtime, is noted down for both client-server and mobile agent based implementations. Mobile agents gives a good performance compared to the traditional client-server model.

IV. CONCLUSION

Mobile agent are better design paradigm for many real life e-commerce application development. Real world entities can be very easily to mobile agents very easily and also mobile agent provide greater flexibility in application design, extendibility and easy integration of new functionalities. Experimental results shows that turnaroundtime Using mobile agents is less compared to client-server model.

REFERENCES

- [1] Danny B. Lange and Mitsuru Oshima, "Seven Good Reasons for Mobile Agents ", Communications of ACM , vol. 42, no. 3, March 1999.
- [2] Alfonso Fuggetta , Gian Pietro Picco and Giovanni Vigna , "Understanding Code Mobility ", IEEE Transactions on Software Engineering , vol. 24(5), 1998.
- [3] Bloomer, J. - Power Programming with RPC, O'Reilly and Associates, 1992.
- [4] P. E. Renaud , Introduction To Client/Server Systems: A Practical Guide For Systems Professionals, Wiley & Sons, 1993.
- [5] J. Dale, A Mobile Agent Architecture to Support Distributed Resource Information Management, thesis, University of Southampton, 1998, 86p.
- [6] Danny B. Lange, "Mobile Objects and Mobile Agents: The Future of Distributed Computing", In Proceedings of The European Conference on Object-Oriented Programming '98, 1998.
- [7] Alfonso Fuggetta, Gian Pietro Picco and Giovanni Vigna , "Understanding Code Mobility ", IEEE Transactions on Software Engineering , vol. 24(5), 1998.
- [8] Concordia Home URL : <http://www.merl.com/HSL/Projects/Concordia/>
- [9] Pattie Maes, Robert H. Guttman and Alexandros G. Moukas, "Agents That Buy and Sell ", Communications of ACM, vol. 42, no. 3, pp. 81 - 91, March 1999.