American International Journal of Research in Science, Technology, Engineering & Mathematics

Available online at http://www.iasir.net

ISSN (Print): 2328-3491, ISSN (Online): 2328-3580, ISSN (CD-ROM): 2328-3629

AIJRSTEM is a refereed, indexed, peer-reviewed, multidisciplinary and open access journal published by International Association of Scientific Innovation and Research (IASIR), USA
(An Association Unifying the Sciences, Engineering, and Applied Research)

DISTRIBUTED INFORMATION RETRIEVAL USING MOBILE AGENT

Prof. Y M Naik, Shilpa Tarihal, Roopali Swami, Ashwini Purandare, Kiran Adike
Department of Computer Science and Engineering,
HiraSugar institute of Technology Nidasoshi-591236
Dist: Belgaum Karnataka,INDIA

Abstract: *Internet has evolved from an information space to a market space with thousands, potentially millions, of electronic storefronts, auctions and other commercial services. This creates great opportunities, but is not without problems.. The information overload is an obstacle to the practical use of potentially useful information on the Web. The use of mobile agents in this kind of applications represents a novel approach and potentially solves most of the problems that exist in centralized client-server solutions, because they are programs with a persistent identity which moves around a network and can communicate with this environment and other agents. We present a possible solution for this problem: the Data Agent system - a mobile agent application for the retrieval of distributed structured information in a scenario of several on-line bookstores. This system was developed for Web-based distributed access to database systems based on Java-based mobile agents.*

Keywords: *Mobile Agents, Distributed Databases, Aglets.*

Mobile agent*: Mobile agents are processes dispatched from source computer to accomplish a specified task each mobile agent is a computation along with its own data and execution state.*
Aglet*: Aglet is a lightweight java object that can move autonomously from one computer host to another for execution, carrying along its program code and state as well as the so far obtained data.*
Distributed databases*: Database which is not found in one particular place, it is distributed everywhere.*

I. INTRODUCTION

In traditional Client-Server way, most of the information retrieval systems do not offer enough flexibility for distributed data repositories. Generally, there are many specifications in traditional way; setting up a connection between the client and server, sending a request to the database server and receiving the result from the server. If there are N servers in the network, the user has to start N network connections and send out N database queries. The following information is needed to transfer on the network:

- Database query request and
- The result data.

The network connection must be maintained all along. When there are more and more mobile devices, the bandwidth is limited and the devices cannot be online always. At that time, the traditional client-server approach cannot fit. As a popular current solution to solve these problems, a distributed and flexible mobile agent-based architecture is proposed. The Distributed Information Retrieval task deals with the collection of information from multiple and usually heterogeneous information sources that exist in a distributed environment.
Mobile agents are agents that can migrate between physical nodes. Mobility is their primary characteristic, not intelligence. Their mobility borrows a lot from process migration. Process migration consists of transferring a process from one computer to another. One of the most common applications for mobile agents is distributed information retrieval where mobile agents can travel to remote information sources, to work locally. Instead of moving large amount of data to the search engine so that it can create search indexes, you dispatch agents to remote information sources, where they locally search indexes that can later be shipped back to the origin. Moreover, the mobile agent can continue the retrieval task even if the network link to the device goes down. Once the link comes back up, the agent sends back its results. Distributed information retrieval using mobile agents makes it easier to design, implement and maintain distributed systems. Mobile agents reduce network traffic and provide an effective means of overcoming network latency. Perhaps most important, through their ability to operate asynchronously and autonomously of the process that created them, they help us to construct highly robust and fault tolerant systems, thereby directly or indirectly benefiting the end user.
Desired functions:

In the Distributed Network Environment, the Mobile Agent performs the following functions:

- **Cloning :** It creates the duplicate copy of itself.
- **Dispatch :** It dispatches the clones to the other hosts in the network.
- **Retrieval :** It retrieves the information from the respective hosts.
- **Processing :** It processes the retrieved information and returns the results back to the origin when the link is operational.

II. RELATED WORK

This work is inspired by current research in many fields. We draw from other agent projects using functionality ideas, various topics in mobility and ideas from distributed-systems research. Software agents have become very popular in the last six or so years. More recently, the capabilities of agents have been applied to electronic commerce, promising a revolution in the way we conduct transactions. A complete summary is returned where the customer can see the prices for all retailers and simply pick an option to be sent directly to the corresponding retailer. This system simply tries to ease the burden for customers to find the best price without manually going around to a number of retailers and compare prices and others systems based on collaborative filtering technology, Firefly helps consumers find products. However, instead of filtering products based on features, Firefly recommends products via an automated "word of mouth" recommendation mechanism called collaborative filtering. The system first compares a shopper's product ratings with those of other shoppers. After identifying the shopper's "nearest neighbors" (i.e., users with similar taste), the system recommends products that neighbors had rated highly but which the shopper may not yet have rated, potentially resulting in serendipitous finds. Essentially, Firefly uses the opinions of like-minded people to offer recommendations. The system is used to recommend commodity products such as music and books, as well as harder to characterize products such as restaurants' web pages. As we've seen, there are several agent based applications that help the user to find the wanted information on the web, but however, we've found a few applications which use mobile agents for distributed database access. Because of this, the main purpose of this paper is to look at how the mobile agent paradigm can improve some distributed database and information retrieval related problems, such as the performance of an ecommerce prototype and stock store market.

III. JAVA MOBILE AGENT TECHNOLOGY

The Aglets Software Developer Kit (ASDK) was developed at IBM Research Laboratory in Japan. It is a framework for programming mobile network agents in Java. From a technical point of view, the IBM's mobile agent called "aglet" (agile applet), is a lightweight Java object that can move autonomously from one computer host to another for execution, carrying along its program code and state as well as the so far obtained data. Unlike an applet's short and boring period of execution, an aglet can exist and execute tasks forever. One of the main differences between an aglet and the simple mobile code of Java applets, is the itinerary that is carried along with the aglet. By having a travel plan, aglets are capable of roaming the Internet collecting information from many places. The itinerary can change dynamically giving the aglet the sense of self-governing and the look of an intelligent agent (that of course is in the hands of the programmer). An aglet can be dispatched to any remote host that supports the Java Virtual Machine. This requires from the remote host to have preinstalled Tahiti, a tiny aglet server program implemented in Java and provided by the Aglet Framework. A running Tahiti server listens to the host's ports for incoming aglets, captures them, and provides them with an aglet context (i.e., an agent execution environment) in which they can run their code from the state that it was halted before they were dispatched. Within its context, an aglet can communicate with other aglets, collect local information and when convenient halt its execution and be dispatched to another host. An aglet can also be cloned or disposed.

IV. PROPOSED WORK

Project Architecture

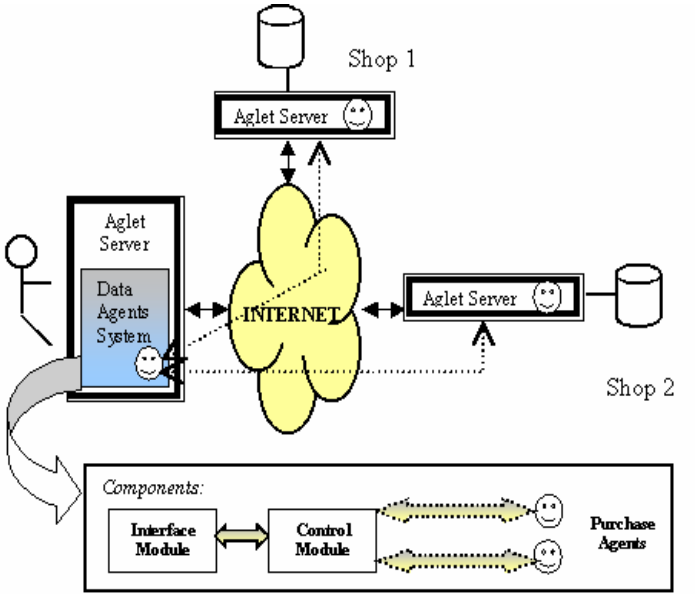
The following architecture is proposed to enable the Data Agents system to have the functionality mentioned above. The system presented comprises the following components: Interface Module, Control Module and Purchase Agents.

Interface Module: this is the component through which the user contacts the system and places his order. This module is also responsible for presenting the result obtained by the group of agents to the user.

"Title", "author", "price range" and "type of itinerary" are the information that the user must provide to the Interface Module so that it may request the Control Module to create and dispatch the purchase agents according to the restrictions imposed by the user.

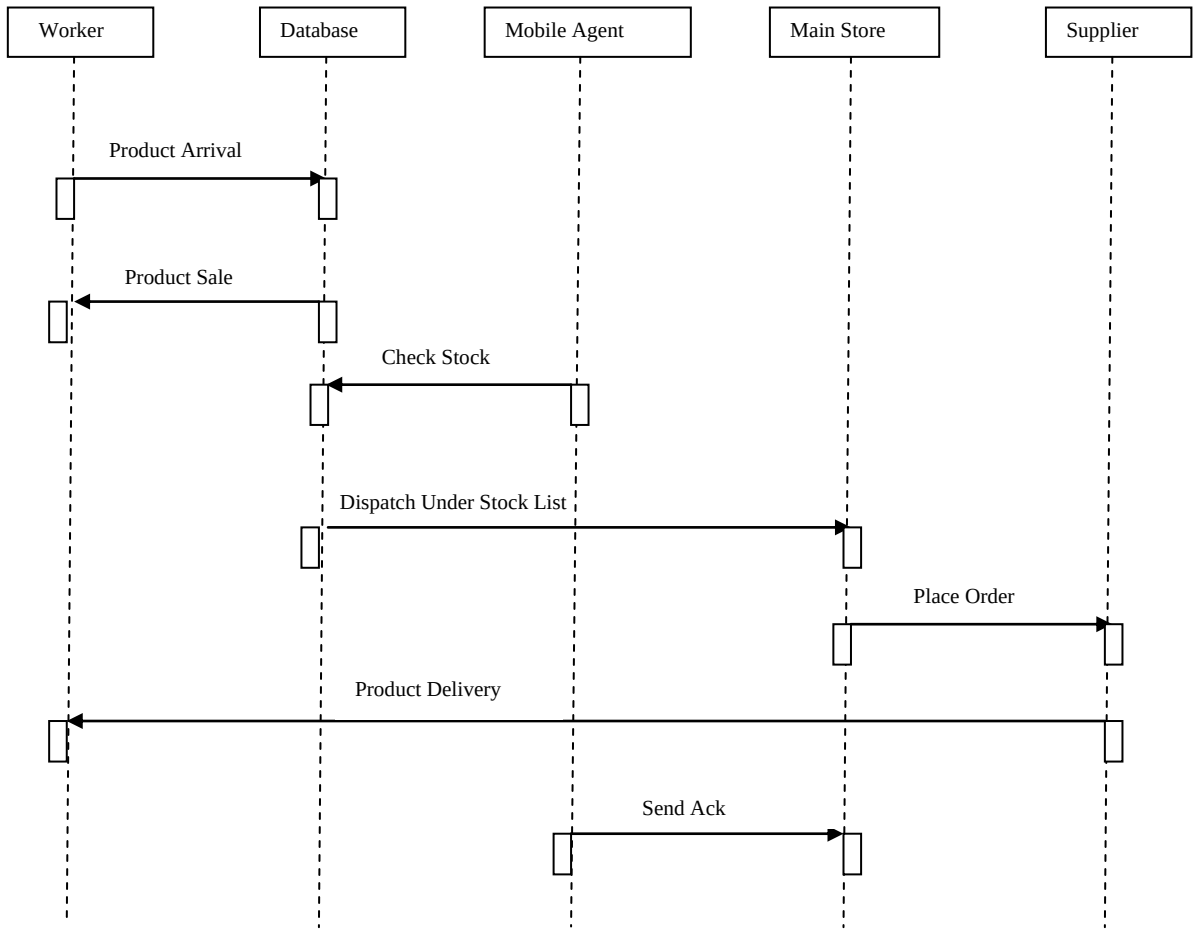
Control Module: This module is responsible for the creation and release of purchase agents to begin the search requested by the buyer. This module also aggregates the results found by the different agents. There is a control module for each type of product available in the system, e.g. a control module for books and a different one for CDs. When the Control Module receives a request to send an agent, the latter is created on the "aglet" layer according to user's requirements and travels through the runtime layer, which converts the agent into an array of

bytes and such array, on its turn, passes on to the ATP layer – Agent Transfer Protocol, to be sent to its destination.



Purchase Agents: Make contact with the stores by accessing their databases, place the order and interpret the answers generated, converting them into a format that is understood by the control module. Before proceeding to their destination, the agents are coded in bit stream: the first segments are general information, such as the agent's identification, and the last segment is the byte array, the agent per se: code and state. The goal of the agents is to check the information found at their destination address, selecting only the information considered relevant and recommended according to the pre-determined rules.

SEQUENCE DIAGRAM



OVERVIEW OF USER INTERFACE

Mainly four components of project main store, sub store worker, mobile agent, supplier

MainStore:

This form will accept username, password and search in database .If found in database Open menu form, which contain sub module links. Allow to make related operations like creating sub stores , grant order placement to supplier ,Check under stock list etc. otherwise do not allow to make related operations.

StoreWorker:

This module accepts username, password of worker. It allows to make following operation like: make stock arrival entry, Sales, Take various reports.

Mobile Agent:

While regular interval. Check the product with under stock criterion [in current interval], Make list, Dispatch mobile agent to main store

There are three possibilities of choices for itineraries:

a. *one agent for each server:*

According to the quantity of servers registered in the system, one agent is created for each server and dispatched to do its task. When each agent arrives at its destiny, it does its search, send the result as a message to Control Module and "dies".

b. *only one agent that visits all the servers:* It is created only one agent that has in its travel plan the addresses of all servers. It will go to all servers, one by one, do the search, send the result as message to Control Module and "dies" at last visited server.

c. *one agent that goes through the servers until to find the first occurrence*: It is created only one agent that contains in its travel plan the addresses of all servers. But it will travel to next server only if doesn't find any book at former server, that is, the agent travels until to find the first occurrence that satisfies the order user.

Supplier:

While regular interval, Check the Order placement, make list, Dispatch Stock to respective sub store.

V. CONCLUSION

In this paper, the mobile agents can dispatch to a destination host carrying the data for computations at remote site. Mobile agents move the data to the remote distributed databases, not the databases to the data. Therefore, the proposed system has huge bandwidth savings and can overcome network latency. Among the many available software agent packages, Java-based aglet package has a few attractive features. Aglets provide a very powerful, simple API allows for quick implementation and easy deployment. The other major involvement of this paper is that a practical prototype has been implemented in the system. In the implementation, it has been developed the searching distributed papers and downloading the required paper. This implementation has been achieved and demonstrated with the technical paper download application from the distributed databases. With the support of this approach, retrieval systems can be built in a truly distributed fashion without the help of a central data repository. Therefore, the failure of a potential single point cannot stop the whole system. So, the users are better supported by using this system.

REFERENCES

[1] Papaioannou, T. On the Structuring of Distributed Systems: The Argument for Mobility. Loughborough University: Doctoral Paper, February 2000.

[2] Chess, D.; et al. Itinerant Agents for Mobile Computing. Journal IEEE Personal Communications, Vol.2, Nº 5, October, 1993

[3] Rosane Maria Martins, Luci Pirmez and Luiz Fernando Rust da Costa Carmo NCE/UFRJ - Núcleo de Computação Eletrônica Universidade Federal do Rio de Janeiro

[4] Mattern, F.; Fünfrocken,S. Mobile Agents as an Architectural Concept for Internet-based Distributed Applications. http://informatik.tudarmstadt.de/VS/Publikationen/papers/kivs99-html/ kivs99.html