International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 2 Issue 5 ǁ May. 2014 ǁ PP.63-67

An Infrastructure Based on a Mobile-Agent for Applications of Ebussiness & Ework

Asst.Prof. Dipayan Kumar Ghosh, Namita Ghosh
Assistant Professor in Computer Science & Engineering (CSE) Depratment Calcutta Institute of Technology(CIT) Uluberia , Howrah – 711316 , West Bengal , India .
MCA, Haldia Institute of Technology(HIT) Midnapore , West Bengal , India.

Abstract: Mobile agents have emerged as a very promising approach for eWork and eBussiness. We have developed an extensive mobile agent infrastructure that supports diverse applications in these fields. Our infrastructure is built around two basic components: a mobile-agent based framework for distributed database access and the PaCMAn (Parallel Computing with Java Mobile Agents) metacomputer. The major functionality of our database framework includes (a) the ability to dynamically create personalized views for the mobile client, (b) dynamic creation and configuration of Web-based warehouses and (c) dynamic support of mobile transactions. PaCMAn offers the necessary tools for Web-based distributed High Performance Computing (HPC) and distributed data mining. Our infrastructure provides the basis for developing eWork applications in many fields. We have utilized it for applications, both wireless and wireline, such as: Electronic commerce, Health Telematics, Teleworking, Distributed Data-mining and Web-based supercomputing.

I. Introduction

Currently we are experiencing tremendous growth in the use of the Internet and the Web. At the same time, we are in the mist of an explosive growth in the use of wireless communications. These two technologies are transforming the way business is contacted. Over the past few years, we have been developing a mobile agent infrastructure that supports business functions on both the Internet and the wireless communication environment. Our infrastructure is built around two basic components: a mobile-agent based framework for distributed database access and the PaCMAn (Parallel Computing with Java Mobile Agents) metacomputer.

Our database framework is based on an extended client/server model and the deployment of mobile agents and provides a new framework for Web-based distributed access to database systems, a fundamental need for eCommerce and virtual enterprises. We have developed a library of database-enabled mobile agents able to support a variety of database functions including: (i) Web database access, (ii) personalized views of databases (iii) a dynamic warehouse infrastructure, and (iv) distributed transactional support.

The PaCMAn (**Pa**rallel **C**omputing with **M**obile **A**ge**n**ts) metacomputer launches multiple Java-mobile agents that communicate and cooperate to solve problems in parallel. Each mobile agent can travel anywhere in the Web to perform its tasks. A number of brokers/load forecasters keep track of the available resources and provide load forecast to the clients. The clients select which servers to use based on the specific resource requirements and the load forecast. We have utilized PaCMAn to harness the, otherwise idle, resources of the Web. PaCMAn provides the basis for Web-based distributed High Performance Computing (HPC) and distributed data mining. We have developed a PaCMAn based Association Rule Mining (ARP) application for Market based analysis and utilized this application to develop association rules for points of sales data from geographically distributed stores.

The remainder of this paper is structured as follows. In Section 2, we present our database framework and in Section 3 the PaCMAn metacomputer along with their deployment in various electronic commerce applications. Section 4 concludes the paper.

II. Mobile Agents for Distributed Information Retrieval

2.1 Mobile Agents for Web Database Access: The DBMS-Agent

Web-based access to databases is an integral part of e-commerce systems and applications. The real challenge is the formation of smart, lightweight, flexible, independent and portable Java DBMS client programs that will support database connectivity over the Internet. The currently proposed methodologies [4] overload the client and offer limited flexibility and scalability. Our approach is based on using mobile agents [1, 8], between the client program and the server machine, to provide database connectivity, processing and communication, and consequently eliminate the overheads of the existing methodologies. Our database framework is comprised of a set of such Java based agents that cooperate to efficiently support Web database connectivity. The main agent, called DBMS-agent, acquires its database capabilities dynamically not at the client but at the server. The other agents of the framework assist this dynamic acquisition.

In particular, in most current approaches to Web database connectivity, an applet at the client machine downloads from the remote SQL server and initiates a JDBC driver, and then handles a complex set of JDBC interfaces. Instead, our proposed DBMS-applet creates and fires a mobile agent (or agents if necessary) that travels directly to the remote SQL server. At the SQL server, the mobile agent initiates a local JDBC driver, connects to the database and performs any queries specified by the sending client. When the mobile agent completes its task at the SQL server, it dispatches itself back to the client machine directly into the DBMS-applet from where it was initially created and fired. Since our mobile agents possess database capabilities, they are called *DBMS-agents*. The DBMS-agent is independent of the various JDBC driver implementations. The DBMS mobile agent can not (and is not supposed to) be aware of which JDBC driver to load when it arrives at an SQL server. Upon arrival at the SQL server's context, the DBMS-agent is informed of all available JDBC drivers and corresponding data sources. The DBMS-agent is then capable of attaching itself to one or more of these vendor data sources.

Our framework promotes a much more efficient way of utilizing the JDBC API and the JDBC driver API and eliminates the overheads of the various conventional approaches. It supports lightweight, portable and autonomous clients as well as operation on slow or expensive networks. The implementation of the framework shows [3] that its performance is comparable to, and in many cases outperforms, the performance of current approaches. In fact, in wireless and dial-up environments and for average size transactions, a client/agent/server adaptation of the framework provides a performance improvement of approximately a factor of ten. For the fixed network, the gains are about 40% and 30% respectively. These performance results were gained while using the Aglets Workbench [5] for the implementation of mobile agents. Since the Aglets Workbench is more tuned towards functionality than performance, we expect our framework to perform even better. This assumption is substantiated by early experiments conducted by other mobile agent technologies (i.e., Voyager, Concordia and Grasshopper) as our implementation platforms.

Furthermore, this new form of Web-based database access supported by the "DBMS-Agent Framework" is [3]: (a) flexible: it can be set up dynamically and efficiently, (b) scalable: its extension to support multidatabase systems not only maintained but also increased its performance benefits, and (c) robust: the statistical analysis performed found the DBMS-agent framework more stable than the current JDBC-based database connectivity. The framework is generic and portable and can be used not only within the Web but stand-alone as well for direct Java database connectivity.

2.2 Dynamic Mobile Agents and Task Handlers

E-commerce system and virtual enterprises are very dynamic by nature. It is also often needed to maintain this infrastructure for a long duration of time but with different responsibilities assigned to the various participants. Participants may switch roles; for example, a site might be a buyer in one transaction, a seller in another and a mediator in a third. This requires a more dynamic and flexible framework. In our framework, we generalize the notion of mobile agent by separating the mobile shell from the specific task code of the target application [6]. We achieve this with the introduction of TaskHandlers, which are Java objects capable of performing the various tasks. TaskHandlers are dynamically assigned to our agents. Each agent can carry with it any number of TaskHandlers and it can dynamically choose when and where to use them.

A TaskHandler Library is a collection of Java objects that are serializable, and thus can travel along with our mobile agents. The TaskHandler is an object that is in the disposal of an agent to use when it needs to perform a specific computational task. For example, when the need arise for a database query, the agent can utilize the DBMS-TaskHandler. The TaskHandlers can be used to dynamically change the role of an agent.

2.3 Views for Wireless and Mobile Clients: Mobile Agents as View Holders

Large-scale Internet queries applied directly to the source databases are often quite expensive. Such queries are even more expensive for wireless clients due to the severe limitations of the wireless links that are in general characterized by high communication costs, limited bandwidth and high latency [7]. The efficient creation and materialization of personalized views provides an attractive alternative. To this end, we have proposed a Dynamic View Framework, DVF, that allows the automatic definition, creation and maintenance of such materialized views.

DVF's multitier architecture is based on mobile agents [10] and is built upon the DBMS-agent Framework. Through DVF, wireless clients identify Web databases of interest, access their metadata, set up dynamically via mobile agents the necessary view infrastructure and create personalized views encapsulated in mobile agents [11] (i.e., the View-agent). The system allows these views to be shared by other mobile clients.

DVF also supports view maintenance in the case of updates at the sources. View maintenance can be achieved in various ways [9]. The proposed infrastructure, via the notion of Task-Handlers, can be dynamically adapted to serve any existing approach. Each agent can carry with it various materialization protocols, as TaskHandlers, and it can dynamically choose (or be directed to) when and where to use them.

DVF brings data close to the clients thus eliminating the overhead of transmitting data over slow networks. In addition, it allows the sharing of views among the clients of the system. Accessing remote views and databases is done both synchronously and asynchronously [2]. With asynchronous communication, clients can pose their query, disconnect and connect later to receive the results. Analogously, busy servers can postpone processing of selected submitted queries. Furthermore, the implementation of the framework through mobile agents allows dynamic code deployment and relocation of views [10, 3].

2.4 Dynamic Warehouse Infrastructure: Mobile Agents as Warehouse Holders

A natural extension of the Dynamic View Framework, (DVF) is the Dynamic Warehouse Framework (DWF). A warehouse can be considered as a collection of materialized views over one or more operational databases. However, this entails many new issues and challenges. New types of mobile agents to deal with these new issues are required. The most vital one is the ***Warehouse_Agent*** that dynamically provides the functionality of a warehouse. From an object oriented perspective one might view the Warehouse_Agent as an extension of the View-Agent of the DVF framework.

The current DWF's implementation gives the ability to its user to create data warehouses over distributed databases, which are query and update independent [12]. Query independence means that every query Q that can be posed to the base relations can be answered in terms of executing a corresponding query Q´ over the warehouse. Update independence means that no queries over base relations are required in order to keep the warehouse data consistent with the base data.

The infrastructure, however, is built in such away that allows the dynamic deployment of any type of data maintenance in much the same way as in the DVF via the notion of Task-Handlers.

2.5 Transactional Mobile Agents

Providing distributed transactional support is crucial for emerging cooperative information systems such as e-commerce, virtual enterprises and the new notion of migrating workflow. These systems operate in a dynamic and distributed environment, dealing with large number of heterogeneous information sources with evolving content and dynamic availability. One feature that distinguishes e-commerce and virtual enterprises is that enterprises can form dynamic partnerships that exist only for as long as they are needed. The stormy, however, explosion of the Internet and the Web found most of these sources unprepared to participate cohesively in distributed cooperative transactions.

Mobile agents provide the ability to dynamically attach the needed transactional functionality to the participating sources. A suite of mobile agents has been developed to support this dynamic transactional infrastructure [13]. The most vital one is the ***TranMan_Agent*** that provides the functionality of a transaction manager. This agent is sent to the resource that will join the virtual consortium to manage the various mobile transactions. Since participation in the virtual team is decided dynamically by the current participants each one of them can direct or clone its TranMan_Agent to the new site as needed.

The TranMan_Agent supports the presumed abort variation of the two-phase commit protocol and timestamping for ensuring the isolation property of a transaction. Note that the transaction tree pertaining to an Internet transaction is set up dynamically. The use of mobile agents can produce an optimized commit tree, namely a flat tree.

III. PaCMAn Metacomputer

The PaCMAn (**Pa**rallel **C**omputing with **M**obile **A**ge**n**ts) metacomputer launches multiple mobile agents that cooperate and communicate to solve problems in parallel. Each agent supports the basic communication and synchronization tasks of the classical parallel worker assuming the role of a process in a parallel processing application. It consists of three major components:

- Broker: It keeps track of the system configuration. All participating clients and servers and their capabilities are registered with the broker. The broker also monitors the systems resources and performs network traffic/load forecasting.
- Server: It is a Java enabled computer (has installed the Java runtime environment) that runs the PaCMAn daemon, in order to host incoming PaCMAn mobile agents.
- Client: It is a Java enabled computer that runs the PaCMAn daemon in order to launch/host PaCMAn mobile agents.

The main responsibility of the PaCMAn mobile agent is to create and coordinate other PaCMAn mobile agents. The PaCMAn mobile agent can assume various roles in the PaCMAn framework the most common are:

- Mobile-Worker (MW): This is the workhorse of the PaCMAn system. It performs all the necessary computation, communication and synchronization tasks. The MW initializes and uses the appropriate TaskHandler.
- Mobile-Coordinator/Dispatcher(MCD): The MCD does not participate in the computation. Instead, it performs task-specific coordination among the mobile-workers. This is done by dynamically assigning tasks to the MWs through the use of TaskHandlers.

3.1 PaCMAn as a Model for Purchasing/Trading Computing Cycles

At any given time, there are a vast number of computers connected to the Internet that are idle. The collective computational power of these computers dwarfs the computational power of all the world's high performance computers put together. At any given time, a client can use all the servers that are registered as available at that time. For example, companies can register their computer as available after workhours on weekdays and for the entire weekends. Home computers on the other hand can register as available during work hours. Of course, users have the ability to make their computer unavailable explicitly by turning off the PaCMAn daemon or implicitly when the local load reaches a predetermined threshold. These are the resources that PaCMAn is targeting. The topology of these loosely coupled resources changes very dynamically.

The PaCMAn model for using computer cycles from idle machines can be effectively used with different business models. For example an accounting system can be setup that supports micro-payments so that any client that uses the system resources will have to pay according to usage. Similarly the account of each server used will be credited with the appropriate payment.

3.2 Data Mining with PaCMAn

Data mining algorithms have extremely large runtimes even for modest amount of data. The proliferation of the Web and the globalization of business have given rise to huge amounts of data been collected and stored in large-scale distributed data warehouses. Thus, the need for efficient data mining is ever more important under the current conditions. The obvious solution is to employ parallel and distributed processing techniques for data mining. In our approach, we go one step further and employ mobile agents that travel to the data sources and communicate and coordinate among themselves to perform data mining. More specifically, we utilize the PaCMAn metacomputer to extract association rules for points of sales data from geographically distributed stores.

3.3 Query Multiple Database Systems

PaCMAn agents are launched to different hosts on the Web, cooperate and communicate with each other to perform complicated tasks efficiently. Thus, PaCMAn agents can be used to query multiple databases in parallel. There are various ways to use PaCMAn agents for querying multiple and possibly heterogeneous database systems. The straightforward way is to create multiple Mobile-Workers (MWs) with DBMS_TaskHandlers (these MW agents are in essence DBMS-agents), load each one of them with an SQL query and dispatch them to the various destinations. The client is responsible for combining the results provided by the various "DBMS-agents".

A more efficient approach is for the client to create an MCD, to whom the responsibility is assigned of creating the multiple "DBMS-agents" (aka MW), loading them with the queries and dispatching them to their destination. The MCD is also responsible for receiving and manipulating the intermediate results provided by the various "DBMS-agents". Only the final result is reported to the client. This approach is in line with our general objective of keeping the client light since the MCD does not necessarily reside at the client. Processing can be done remotely with only the final result transmitted to the client.

IV. Conclusions

Two key technologies, the Internet/Web and mobile communications, are transforming the way we contact business. Over the past few years, there is a tremendous effort in developing appropriate business models and technologies that will utilize these new technologies for business. Most of the current approaches work well with only one of the two. E-commerce systems and virtual enterprises have a very dynamic nature; they often need to form dynamic partnerships that exist only for as long as they are needed. This is a crucial requirement that the current approaches fail to effectively fulfill. Mobile agents provide an efficient platform for distributed dynamic processing which works equally well with both the fixed and wireless networks.

We have developed a mobile agent infrastructure that supports many of the functionalities that are needed for eWork and eCommerce. Our infrastructure provides support for distributed information retrieval: database queries on single and multiple databases, personalized views, dynamic data warehousing and transaction processing. PaCMAn is a Web-based metacomputer that supports collaborative processing. PaCMAn can be utilized for eBussiness functions such as queries to multiple databases and distributed data

mining. Furthermore, PaCMAn itself provides a model for trading idle computer cycles. Finally, our mobile agent infrastructure is heavily dynamic thus effectively serving the dynamic nature of E-commerce systems and virtual enterprises

References

[1] Colin G. Harrison, David M. Chessm, Aaron Kershenbaum. Mobile Agents: are they a good idea? Research Report, IBM Research Division.

[2] Panos K. Chrysanthis, Sujata Banerjee, Shi-Kuo Chang, "Establishing Virtual Enterprises by means of Mobile Agents", RIDE Workshop, Sidney, March, 1999.

[3] S. Papastavrou, G. Samaras, and E. Pitoura, "*Mobile Agents for WWW Distributed Database Access",* Proc. 15th International Data Engineering Conference (IEEE-ICDE´99) (Best Paper Award), Sydney, Australia, March 1999.

[4] Roedy Greem. Article: Java Access to SQL Databases. Canadian Mind Products, 1997.

[5] Aglets Workbench, by IBM Japan Research Group. Web site: <http:/ aglets.trl.ibm.co.jp>.

[6] P. Evripidou, G. Samaras, E. Pitoura and C. Panayiotou*, "PacMan: Parallel Computing Using Java Mobile Agents*", 13th ACM International Conference on Supercomputing (ICS), Workshop on "Java for High Performance Computing"**,** Rhodes, Greece, June 1999**.**

[7] E. Pitoura and G. Samaras, "*Data Management for Mobile Computing"*, Kluwer Academic Publishers, ISBN 0-7923-8053-3, 1998.

[8] Nick Roussopoulos: Materialized Views and Data Warehouses. KRDB 1997: 12.1-12.6

[9] Constantinos Spyrou, George Samaras, Evaggelia Pitoura, Evripidou Paraskevas "*Mobile Agents for Wireless Computing: The Convergence of Wireless Computational Models with Mobile-Agent Technologies***",** Journal of ACM/Baltzer Mobile Networking and Applications (MONET), special issue on "Mobility in Databases & Distributed Systems", 2000. (in print).

[10] Constantinos Spyrou. "*A Library of Java Mobile Agents for Distributed Database Access*." Undergraduate thesis supervised by George Samaras, Computer Science Department, University of Cyprus, June 1999.

[11] D. Laurent, N. Spyratos, et.al, "Complements for Data Warehouses", Proc. 15th International Data Engineering Conference (IEEE-ICDE´99), Sydney, Australia, March 1999.

[12] Evgenios Charalampous. "*Mobile Agents for Distributed Transactions*." Undergraduate thesis supervised by George Samaras, Computer Science Department, University of Cyprus, May 2000.