Table of Contents Search Author Index

An Intelligent Approach for Integrity of Heterogeneous and Distributed Databases Systems based on Mobile Agents

M. Anber and O. Badawy

Department of Computer Engineering,
Arab Academy for Science and Technology and Maritime Transport,
Alexandria, Egypt.
P.O. Box 1029, Alexandria, Egypt.

Emails: mahdianber@hotmail.com, obadawy@aast.edu

Abstract

In today's networked world, information is widely distributed across many independent databases in heterogeneous formats. Integrating such information is a difficult task and has been addressed by several solutions. However, previous integration solutions, such as the conventional, have several shortcomings. Database contents and structure change frequently, and users often have incomplete information about the data content and structure of the databases they use. When information from several such insufficiently described sources is to be extracted and integrated, two main problems have to be solved, how can discover the structure and contents of and interrelationships among unknown databases, and how can provide durable integration views over several such databases. This paper developed solution for those key problems in information integration. This paper addressed Mobile Agent for Heterogeneous Distributed Databases (MAHDD) system. MAHDD is the system which proposed and implemented in this paper. The aim of this system is to solve the heterogeneity problems among multiple distributed databases. The aim is actually achieved by apply the MAHDD system on a maritime prototype, this prototype includes three heterogeneous databases and each one of them is in different location and simulate different enterprise. MAHDD system achieved the integration for these three databases in this prototype, and it gave the users the ability to pose a single query which deal with these heterogeneous distributed databases as a single database. The implementation of MAHDD system on this prototype followed hierarchical approach. This approach can be considered to assure the robust of the data flow during runtime. The important excellences of MAHDD system that, it is an intelligent system; it used a lot of Meta-Knowledge to control dataflow during run time, to fulfill its duties during shorter time, and to achieve high performance with easy administrate.

1 Introduction

Mobile agents have a unique ability to transport themselves from one system in a network to another. The ability to travel allows mobile agents to move to a system that contains services they want to interact with and then to take an advantage of being in the same host or network as the service [6]. Bringing new promising possibilities for distributed systems development, mobile agent technology is attractive especially in information retrieval in widely distributed heterogeneous environment, network management, electronic commerce and mobile computing. Upon some basic distributed computing models adapted to the mobile agent paradigm, a set of frameworks for distributed database access over the World Wide Web were proposed and tested [1,3, 4, 5].

MAHDD is a system that integrates a set of heterogeneous databases under one virtual global scheme and transparently manages distributed execution of user's global requests, [1]. System services can be accessed via HTML in web browser. The system gives the user the possibility of exploit data stored in multiple remote databases in the same manner as if they are stored in centralized database system. So, the user is shielded from all the complexity of distribution and heterogeneity. The essential features of this system ***distribution***, ***autonomy*** and ***heterogeneity*** of underlying databases are the crucial issues that determine high complexity of the global system, where as distribution refers to the distribution of data, autonomy deals with distribution of control, and heterogeneity can affect data models, access mechanisms, or other platform specific issues.

It is important to note at this point that the mobile agent technology can successfully solve the contradiction between efficient execution and easy administration in multidatabase systems.

First of all, installing agent servers providing uniform database access on the database machines that can hide underlying heterogeneity while preserving database autonomy. Secondly, easy administration can be achieved by the fact that local changes maybe done by incoming agents and no additional local access are required. Finally, agent servers and agents can communicate to each other in the sense of fully interconnected network topology, so, the advantage of optimal data transfer during global execution can be exploited. Today's the most utilize model for distributed sources access is the client-sever model that forms two-tier (or generally multi-tier) architecture. The client-server paradigm strictly divides the distributed application code into the server part providing fixed services at the server machine and the client part remotely requesting them, [7]. The mobile agent technology provides an alternative way of distributed programming. The mobile agent technology is based on the idea of process (or thread) migration together with its program code and its execution state to different hosts during the execution time, [7]. This transferable unit is called mobile agent. From the user's point of view, mobile agents are computer programs that can be dispatched together with their data from a client computer to the most beneficial host for remote execution. The decision to dispatch can be done by the agent itself or can be initiated by other agents or an application. MAHDD system is technically based on java technology which is widely considered as a powerful application development tool in mobile computing. MAHDD system has used Java-based mobile agents implemented in Aglet Software Development Kit ASDK [1, 8]. This paper is organized into sections. Section 2 presents MAHDD system architecture.

Top

Section 3 presents mobile Worker agent’s plan. Section 4 presents MAHDD system implementation and testing. Section 5 presents performance evaluation. Finally, section 6 reports the conclusion with future work.

2 MAHDD System Architecture

A general architecture and methodology have been illustrated and implemented in MAHDD system is based on software mobile agent as show in Figure 1. The proposed idea depends on using mobile agents between the remote clients and databases servers’ machines, [1].

- The proposed solution creates and fires a mobile agent (or agents if necessary) that travels directly to the remote Databases (DB) servers. At each DB server, each mobile agent connects to database and performs any queries specified by the sending client. When mobile agent completes its task at the DB server, it sends the results of execution queries back to the client machine through web server. The functionality of MAHDD system is accomplished by mobile and stationary agents that communicate to one another via messages.

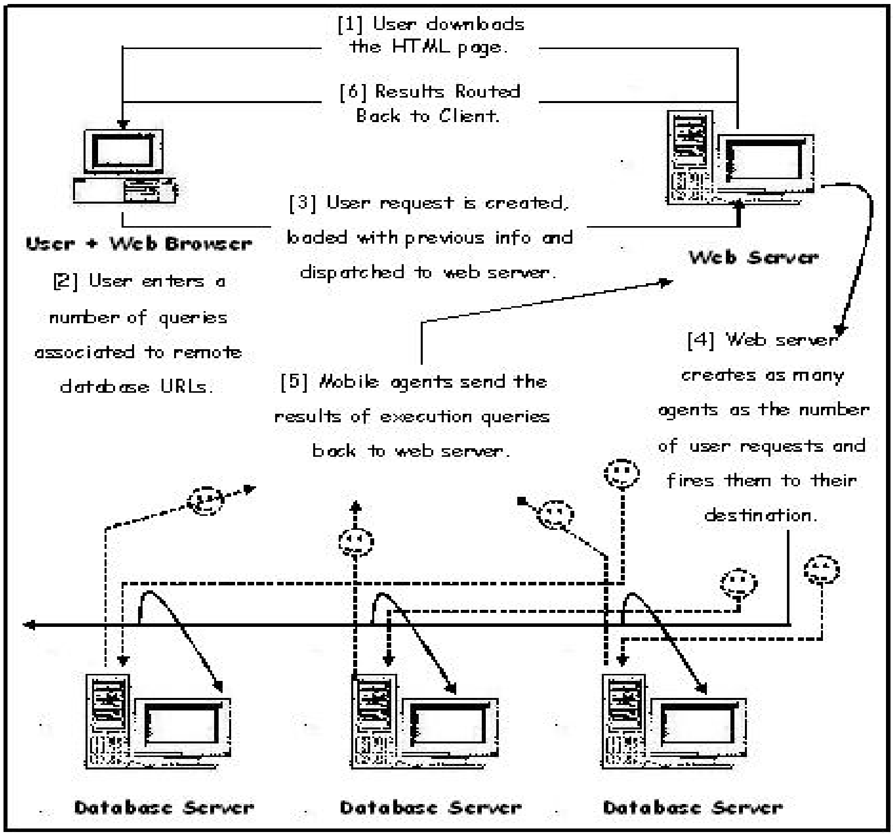
[1] User downloads the HTML page.
[6] Results Routed Back to Client.
User + Web Browser
[3] User request is created, loaded with previous info and dispatched to web server.
Web Server
[2] User enters a number of queries associated to remote database URLs.
[4] Web server creates as many agents as the number of user requests and fires them to their destination.
[5] Mobile agents send the results of execution queries back to web server.
Database Server
Database Server
Database Server

Figure 1: Mobile Agent in MAHDD Architecture.

Top

Therefore, each of functional units, which can be identified in the system, must run an agent server to provide agents with the runtime environment. In MAHDD system implementation, used Tahiti agent servers delivered with ASDK packages. MAHDD system comprises functional units of two types:

- The **Central Unit** forming the core of the system,
- Multiple **Workplaces** units serving as database access points.

There is just one **Central Unit** in the system. The Central Unit acts as a distributed database server. It stores information about the system, accepts user's requests, starts and manages their execution. Additionally, the Central Unit usually stores the program codes of all agents regardless the site on which they will operate later.

The **Workplaces** are the ports from which incoming agents access the databases through the Java Database Connectivity JDBC that is the borderline between uniformly designed global system and possibly heterogeneous databases. A Workplace can connect one or more databases. To establish a Workplace, just a general agent server with appropriate configuration parameters and a proper JDBC driver must be installed at that site.

The additional Workplace's functional codes come as agents. The only inevitable information, which the global system must hold to be able to work with database, is the URL of the related Workplace. The logical concept of functional units does not require their distribution to different sites. If advantageous, several Workplaces can run on the same machine. Moreover, it is possible to combine the Central Unit and one of the Workplaces such that both of them run the same agent server, Figure 2.

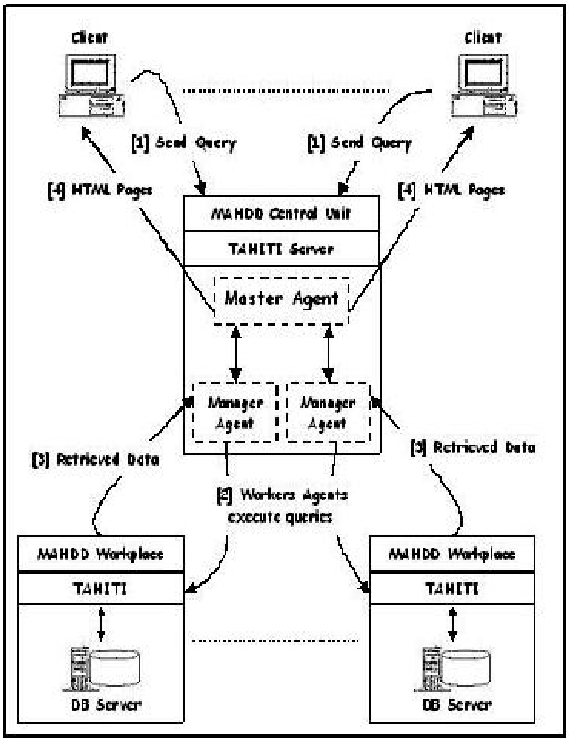

Figure 2: General Architecture of MAHDD.

Top

The following are the basic types of agents operating on functional units and short description of their responsibility:

1. The **Master** stationary agent is the organization chief of the system. There is just one agent of this type. It is located at the Central Unit. The Master takes care of global information of the system, manages joining and disjoining remote databases through Workplaces, establishes user sessions, and provides an http gate to the system. Master agent contains a lot of Meta-Knowledge, which made it an intelligent agent and gave it the controlling of data flow during runtime. This Meta-Knowledge gave the Master agent the ability to create the suitable Manager agent for suitable user request.

2. A **Manager** stationary agent is created by the Master at the Central Unit to handle one user's session. The main role of the Manager is to create Workers agents and to execute individual user's requests. Additionally, the Manager coordinates global transactions. Manager agent contains Meta-Knowledge which guides it, when should create the Worker agent, and how number of Worker agents it should create.

3. A **Worker** mobile agent is designed to perform databases operations. Workers usually created by a Manager to carry out one user's request. Created at the Central Unit and equipped with mission information, a Worker migrates to the proper Workplace, and accomplished its database tasks. Meta-Knowledge inside Worker agent guides it to know which route it will follow. Meta-Knowledge plays the most important role in the Worker agent to guide it when and how it can connect to database and which driver it will use.

3 Mobile Worker Agent's Plan

There are three strategies for mobile Worker agent, and all of these strategies are implemented through the testing of the MAHDD system, [1].

1. Single Worker Agent (many Dispatching and Retracting),
2. Multi Worker Agents,
3. Roaming Worker Agent.

4 MAHDD System Implementation and Testing

This section clarifies a brief portion about how MAHDD system has implemented and tested. It clarifies the tools which MAHDD system has used in the implementation.

4.1 Tools which Used in MAHDD System

MAHDD implemented using Java Development Kit (JDK 1.3) and Aglets Software Development Kit (ASDK ver. 2.0.2) from IBM, [6, 8]. ASDK, one of the well-known agent systems, is a set of java packages that enable users to deploy and manage their own

Top

mobile agents and agent servers. It provided an easy installation, intuitive and flexible deployment, comfortable runtime environment (Tahiti and Hawaii agent server). The testing of the system was on databases servers of different vendors on different platforms: Oracle 8i enterprise edition, SQL Server 7.0 and Microsoft Access 2000 running on Windows 2000 Advanced Server and on Windows 2000 Server.

4.2 Prototype Heterogeneity Level

1. Database Management System (DBMS) Level: this paper has used three relational DBMS, and each one has its own entities, relations, contents and structure.
2. Semantic Level: there is a semantic conflict between the entities representation in the three databases.
3. Platform Level: there are two operating systems, windows 2000 advanced server and windows 2000 server.

5 Performance Evaluations

The performance evaluation computes the total time required by a local user (intranet) or a global user (internet) to access and query heterogeneous and distributed databases. Data manipulation techniques in the web application depend on the web server processor. The performance and the throughput of retrieving data objects are not only depending on the web server's configuration but they depend on the internet traffic.

In this application exercise, the process of retrieving data has been examined under two environments.

One is through a personal web server that runs under an 800 Pentium III processor at 128 MB of RAM and without the connection with the internet. The other is through the internet with the same configuration and it has a real internet protocol address, (internet speed was about 28 to 34 kb/sec).

5.1 Retrieval Data through Intranet

The test is running on three distributed and heterogeneous databases, follows the roaming Worker agent's plan. The number of records is changeable from experiment to other and the transfer and processing time is changed as result to the changing in number of records. Figure 3 shows the results of this experiment. This figure shows that there is a cumulative relation between number of records and transfer and processing time.

5.2 Retrieval Data through Internet

These experiments are a testing through internet, and they are stress testing from three regions (Canada, Australia, and Jordan).

5.2.1 Test 1

This test is running on two heterogeneous and distributed databases, follows the roaming Worker agent plan and the number of records is changeable from experiment to other and the transfer and processing time is changed as the result to the changing in number of

Top

records. This test is performed between web client in Canada and distributed servers in Egypt.

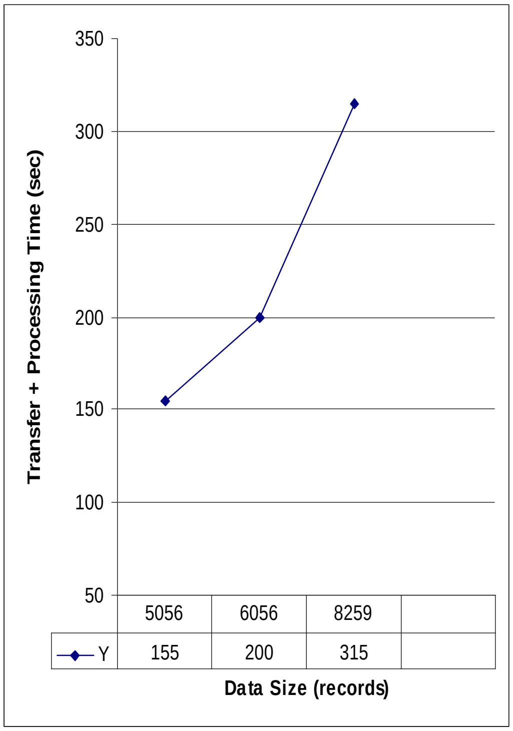

Figure 3: Testing through Intranet.

Figure 4 shows the results of this experiment. This figure clarify that there is a cumulative relation between number of records and transfer and processing time.

5.2.2 Test 2

The test is running on three databases, follows the roaming Worker agents plan and the number of records are 8259 record during this experiment. The transfer and processing time in this experiment was 390 to 405 second. This test is performed between web client in Canada and distributed servers in Egypt.

All the last experiments have presented the transfer time through network and processing time in distributed servers, which mean the time interval between sending queries from user site (whether user is a local user or global user), and receiving the results. It is important to mention here that there is no lost in any record of the results. The user receives all the records without any losing.

Top

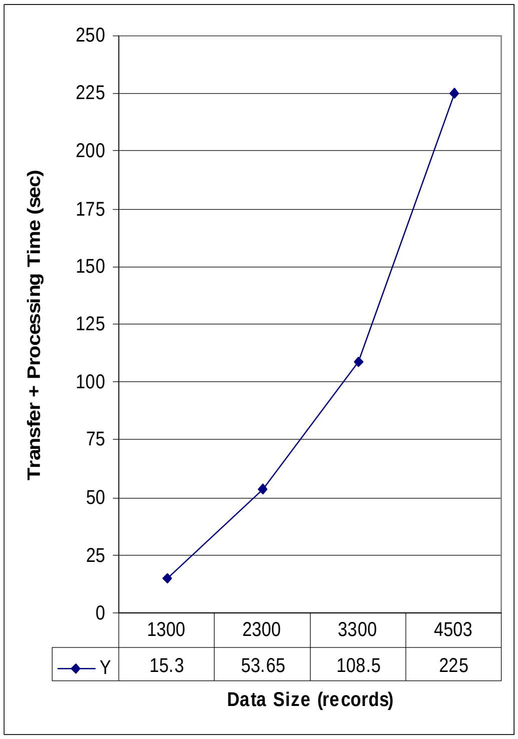

Figure 4: Tes1 through Internet.

6. Conclusion and Future Work

In this paper, we have presented new system called MAHDD system. This paper has presented how a system for heterogeneous and distributed databases integration can be improved by the exploitation of mobile agents.

In particular, agents are useful in the management of the sources, which, in an open and dynamic scenario like the internet, can be spread and can change in an uncontrolled way. There is no integrated schema and component schema in MAHDD system. In this system, there is no any pre-installed code in any workplace site. MAHDD system can make update, insert and retrieve data and it can deal with unlimited user requests. This system save bandwidth and reduce network load by using mobile agents paradigm.

MAHDD system reveals two important features; first, heterogeneity of individual databases can be hidden under the uniformly designed Workplaces, second, the right integration code comes to Workplaces as mobile agents when needed instead of pre-installing it locally.

Top

Hierarchical approach is considered in MAHDD system to assure the robust of the data flow during the runtime. MAHDD system can control the data flow by putting large amount of Meta-Knowledge in Master agent. Meta-Knowledge gave the system the ability to be, intelligent system, and to control and observe the movement of mobile agents when Central Unit creates these mobile agents and dispatches them to their destinations (Workplaces). By applying hierarchical approach using Meta-Knowledge, the system can save the time and provide high performance and reliability.

With regard to future work, this paper sketches some research directions. The first one relates to the dynamic integration of Heterogeneous Distributed Database Systems HDDBSs by exploiting mobile agents that search for interesting new databases. The second one is looking to design on-line mobile agents that can teach themselves a new expertise and information at runtime. The third one is looking to design mediator agents which are as a mediator between the visitors' agents and database. The forth one is the interaction between agents, which can occur in different ways and with different languages. Then, an area that is worth to be explored is a security issue, it is still another problem in this research field and it is very important, [2].

Top

References

[1] Anber M., (July-2003), An Intelligent Approach for Integrity of Heterogeneous and Distributed Databases Systems based on Mobile Agents, Master of Science, Computer Engineering. Arab Academy for Science and Technology and Maritime Transport.

[2] Claessens J., Preneel B., and Vandewalle J., (2003), How Can Mobile Agents Do Secure Electronic Transactions in Untrusted Host, A Survey of the Security Issues and the Current Solutions, in ACM Transactions on Internet Technology, Vol. 3, No. 1.

[3] Anber M., Badawy O., (2002), Integrating Approaches for Heterogeneous and Distributed Information Sources, in 12th International Conference on Computer Theory and Applications, Alex, Egypt.

[4] Papastavrou S., Samaras G., and Pitoura E., (2000), Mobile Agents for WWW Distributed Database Access, In Proceedings of the 15th International Conference on Data Engineering, pages 228-237. IEEE.

[5] Samaras G., Dikaiakos M., Spyrous C., and Liverdos A., (1999), Mobile Agents Platforms for Web Databases: A Qualitative and Quantitative Assessment, Appeared in Proceedings of ASAMA, pp.50-64.

[6] Lange D.B., Oshima M., (1998), Programming and Deploying Java Mobile Agents with Aglets", Addison-Wesley.

[7] Lange D.B., (1998), Mobile Objects and Mobile Agents: The Future of Distributed Computing, Springer-Verlag, Berlin Heidelberg.

[8] Aglets SDK:
URL: www.trl.ibm.co.jp/aglets/

Top