IJFEAT

INTERNATIONAL JOURNAL FOR ENGINEERING APPLICATIONS AND TECHNOLOGY

Operational Model with DBMS-Aglet

Ms. Bhagyashree Patle[*1], Ms. Manasi Kulkarni[2]

[1]*Student, Computer Science and technology, VJTI Mumbai, Maharashtra, India,*

bhagyashreepatle@gmail.com

[2]*Professor, Computer Science and technology, VJTI Mumbai, Maharashtra, India,*

mukulkarni@gmail.com

Abstract

Now a day's web becomes a more popular for accessing information from the remote server. User requires different information at a time on same web pages; this is the main issue in writing this paper. This paper explains how to provide different information at a time to user. How to make system scalable, reliable and flexible?

Index Terms: *Mobile Agent, Aglet, JDBS, DBMS-aglet, Applet.*

1. INTRODUCTION

The java is frequently used because it provides mainly portability and security control system. It also helps in to communicate client and server. Web browser provides end user interface and database management system for large application.

Now we have to combine these two technologies, namely java and web for retrieving information from the database. The main challenge is make flexible, independent and portable Java DBMS client program which will support database connectivity. We use mobile agent between client and server for the database connectivity, communication and reducing the overhead.

We refer the framework called as "DBMS AGLET FRAMEWORK". The main agent in this framework is the DBMS-aglet which acquires database capability at server side not at client side.

2. BACKGROUND

2.1 Required Technology

JAVA and JDBC:

The uniqueness of Java is it combines both compiled and interpreted code. Java bytecode represents the instructions for a virtual microprocessor called java virtual machine (JVM). It is also known as Java interpreter.

Java applet is a java object program that execute within the Java enable web browser. Restrictions of Java applet are-

1. It is not allowed to access any local resources of the client.
2. It is not allowed to communicate with URL other than they were downloaded from.

Java Database Standard Connectivity (JDBC) is used for accessing and manipulating relational database. It consist of two layers 1) JDNC API , it provide interface for relational database. 2)JDBC driver API , it is used to execute interface.

Mobile Agents

Mobile agents are processes which are dispatches from the client to complete specific task. When the mobile agent is reached at server then it sends to the agent execution environment. The mobile agent transport itself to another server for accomplishing a task.

The IBM's mobile agent is called "Aglet". It is light weight java object which transfer one host to another autonomously.

The aglet can be dispatched to only that host which support JVM. It requires the remote host should preinstalled Tahiti. Tahiti server listen the port no of incoming aglet and provide aglet execution environment.

2.2 Applets for Distributed Database Access:

There are four ways to access remote host through Java applet. JDBC-Bridge Driver: it translates the JDBC API calls into ODBC calls and sends them to the ODBC driver. JDBC Native driver: it uses JDBC driver written half in Java and half in the native code.

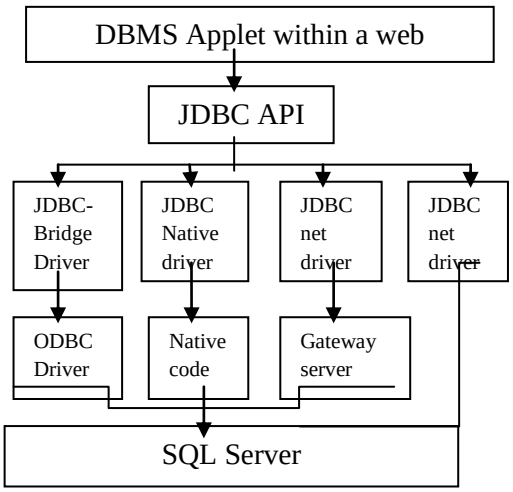


Fig1. JDBC Approach

1. JDBC net driver: The client applet, through this JDBC driver speaks an intermediate language that is translated by a middle-tier gateway at the server into a DBMS-specific protocol and eventually passed to the SQL server.
2. JDBC net driver: It is also written entirely in Java. The driver can be fully downloaded to the client applet and speaks a DBMS-vendor protocol directly to the remote SQL server.

3. DBMS-AGLET FOR DISTRIBUTED DATABASE

DBMS-Aglets create and fire a mobile agent that travels directly to a 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.

3.1DBMS Aglet Framework

Applets need to be enhanced to provide database specific interfaces and be capable to host agents with database capabilities. In addition, the existing aglets need to be extended to be database capable.

1. **DBMS-applet**: responsible for forming a graphical client database interface that the user can utilize to input database requests.
2. **DBMS-aglet**: responsible for carrying the user's request directly to the remote database, executing it, and returning the results back to the DBMS-applet context.
3. **DBMS-assistant stationary aglet:** Its responsibility is to inform any incoming DBMS-aglets carrying database requests about the available JDBC drivers and data sources and to assist them in carrying out their requests.

3.2Description of Web Based DBMS-Aglet Framework:

DBMS-applet and a DBMS-aglet are included in an html page at the web server machine. Install an aglet router at the web server machine and a Tahiti aglet server at the SQL server Machine.

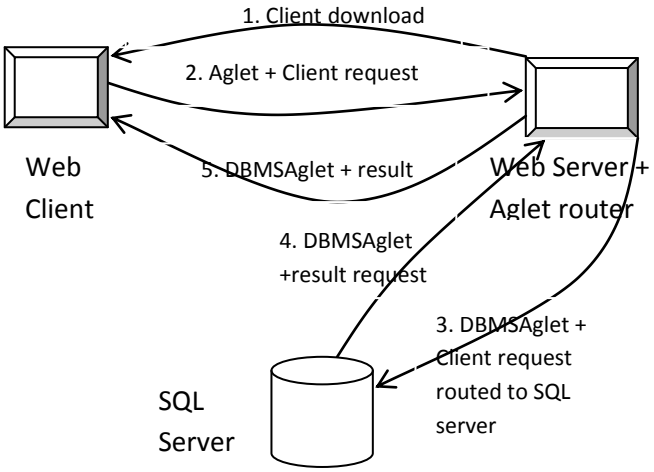


Fig2. Routing Request trough DBMS Aglets

Execution begins with a web user downloading into his Java-enabled browser an html page that contains a DBMS-applet and the DBMS mobile agent.

Then, the DBMS-applet dispatches the aglet from its context to the URL of the SQL server. Due to Java applet security restrictions mentioned earlier, the aglet must first go through the web server URL from where the html page containing the DBMS-aglet was downloaded.

At the Web server machine, an aglet router captures the incoming DBMS-aglet and immediately forwards it to its destination.

Arriving at the SQL server machine, the DBMS-aglet is received by a Tahiti aglet server. The DBMS-aglet communicates with the DBMS-assistant stationary aglet to be informed of the available JDBC drivers to load, and the available data sources to get connected to.

When the aglet completes its task at the SQL server, it can either dispatch itself to another SQL server to perform another user's query or it can dispatch itself back to the DBMS-applet at the client machine, going through the aglet router at the web server machine.

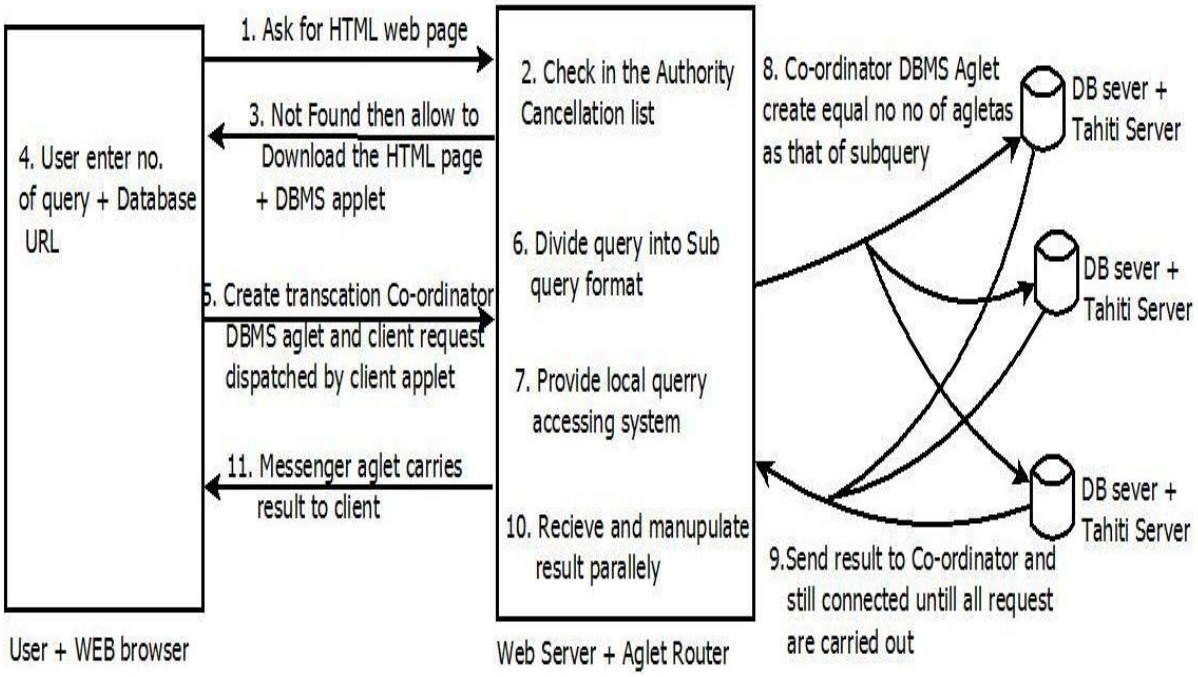


Fig3. Operational Model for "Mobile aglet Framework"

3.3Refining the DBMS aglet framework

The role of aglet is to send various database requests to the SQL server and bring back the result. In this way the client is relaxed from responsibility of downloading JDBC drivers. The request is issued and aglet fire to the database every time increases the overhead.

3.3.1 DBMS-Aglets and messenger Aglet

To eliminate this overhead, extend client server model into three tires called client-agent-server model. Agent is placed between the paths of client to server. Now any communication is executed through this agent.

Whenever client want to access remote SQL server, he release now two aglets one is parked DBMS aglet which load the appropriate JDBC driver, connect to database, submit the request and collect the filtered information.

Another aglet is messenger aglet; its responsibility is to carry result back to the client. This is very efficient approach for large amount of request is come to the same server. If the parked DBMS-aglet moves to another server the messenger can deterministically follow it.

3.3.2 DBMS-Aglets using messages

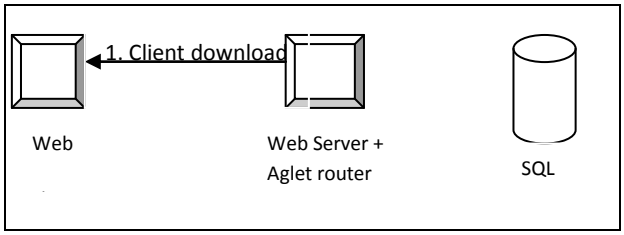
The parked DBMS aglet attached locally to the database server, it eliminate one overhead that is time to reload JDBC but there is one more overhead is remains that is time required to send reply back to the client.

To eliminate those overhead the messenger are further splits into two messages. The first message is delivered from the parked DBMS-aglet to the DBMS applet and contains the results of the last query. The other message from the
DBMS applet to the DBMS-aglet contains the new client query and any additional directions to the parked DBMS aglet that might be needed.

4. PROPOSE WORK

In this paper we studied traditional approach and current approach for accessing remote SQL server through various aglets. Now we propose how this will operate exactly in the transaction processing. For this we design an operational model which shown in fig 3.

In this model, user wants to do transaction on the distributed database system. First he will request for Web page when request is at web server, it will check the ACL (Authority Cancellation list) (step 2). If there no match found then allow to download the HTML page along with the DBMS applet (step3). Otherwise access is denied.



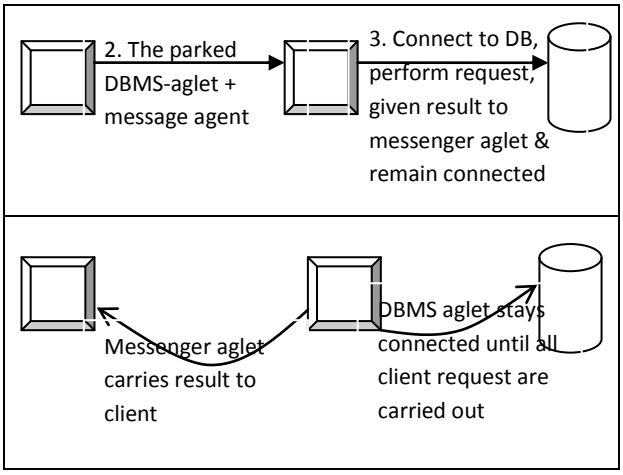


Fig4. DBMS-aglet framework using a messenger aglet

After getting DBMS applet user enter number of query (step4). The DBMS applet fires the query in the form of applet. Now the messenger travels these queries to web server and create a coordinator (step 5). The queries are comes from the various heterogeneous system web server have to convert it into sub query and provide local query processing system (step 6&7).

Coordinator DBMS aglet create number of aglets equal to sub query generated and send them to their desired destination (step8). DBMS aglet is received by a Tahiti aglet server. The DBMS-aglet communicates with the DBMS-assistant stationary aglet. The DBMS-aglet connects to the appropriate SQL server to execute the query.

Aglet completes its task at SQL server or dispatch itself to other server to solve the user's other request. After completing request it sends to the web server (step 9). Web server receives the result and manipulates it simultaneously. The result is send back to the client which initiates the query through the messenger.

CONCLUSION

From the above frame work we make the client applet light and portable. This frame work make client applet free from-

- Downloading JDBC drivers
- Not using any JDBC API classes at the client's DBMS-applet

The only responsibility for the client is to specify the URL address of the database server.

The DBMS mobile agent cannot be aware of which JDBC driver to load when it arrives at an SQL server.

REFERENCE

[1] Stavros Papastavrou, George Samaras and Evaggelia Pitoura "Mobile Agents for World Wide Web Distributed Database Access" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 12, NO. 5, SEPTEMBER/OCTOBER 2000

[2] Jihoon Yang; V. Honavar ; L. Miller ; J. Wong, "Intelligent mobile agents for information retrieval and knowledge discovery from distributed data and knowledge sources", IEEE Information Technology Conference, Page(s) 99-102, 1998.

[3] Bhavani Thuraisingham, Member, IEEE Computer Society, and William Ford, "Security Constraint Processing in a Multilevel Secure Distributed Database Management System" IEEE transactions on knowledge and data engineering, vol. 1, no. 2, APRIL 1995