Web-Based Database Distributed Systems

Assist. Alexandru Dan CĂPRIȚĂ, Assoc. Prof. Vasile MAZILESCU, PhD
Department of Accounting and Economic Informatics, University "Dunărea de Jos" Galați

Distributed systems have evolved from traditional "LAN" based distributed systems towards "Internet based" systems. This article outlines Web-based distributed database main features and architectures, as a key component of many Internet related applications, such as applications in electronic commerce, information retrieval, and multimedia. Finally, we present a generation 2 WBDB application for students' activity management.
***Keywords**: distributed systems, Web-based databases, Web applications.*

1 Introduction

World Wide Web has changed the way we do business and research. It also brings a lot of challenges, such as almost infinite content, resource diversity, and maintenance and update of contents. Web-based database (WBDB) represents one of the answers to these challenges.

A **distributed system** is a system consisting of a collection of autonomous machines connected by communication networks and equipped with software systems designed to produce an integrated and consistent computing environment [2]. Distributed systems enable people to cooperate and coordinate their activities more effectively and efficiently. The key purposes of the distributed systems can be represented by: resource sharing, openness, concurrency, scalability, fault-tolerance and transparency.

A **web-based database** is a database that resides entirely on an Internet server. Access to the database is through a web browser and usually utilizes a password system that allows for restricted access to users depending on the privileges they have been given.

WBDB can be used for a range of functions:

- Creation of product catalogues;
- A back end for e-commerce allowing for instant update of prices, product details etc;
- Maintenance of client or user details for email, reference etc.
- Frequently updateable newsletters, company activities, meetings etc;

WBDBs possess a number of advantages:

- ***Maintenance and updating*** - a WBDB separates content (database) from presentation (an HTML page). It means that the owner of a site is able to update the content of the site without constantly having to go through its webmaster or designer. Creating a Web template once and merging it with new content (database) is a more reliable way than publishing information with a consistent layout;
- ***Reusability and modularity*** - by designing additional templates, one can easily reuse content on another Web site or modify it to fit a new design. For users, databases make site searches more accurate: they can be limited to certain fields, returning better-quality hits than full-text searches;
- ***Distribution of data update*** - with the right interface, even a novice user can update database information; the Web publishing system can then send out the changes;
- ***Security*** - databases help ensure that contents are accessed by authorized users.

WBDB is JUST in time, and already works in many fields. The researchers' tasks are to make it evolve rapidly and satisfy the user's requirement by developing new methods, languages, and frameworks. In Table 1 are present some features available for WBDBs:

Common features	New functions
Password access and privilege-based restrictions;	Keeping track of the origin and modification history of each article by the use of a database management systems (DBMS);
Ability to download database files as text or tab delimited files that can be read by a database or spreadsheet program on the local computer;	Obtaining valuable new data by tracking and logging user activity and user contribution in the process of interaction;
Ability to include images, email links and hyperlinks to other web pages in the database output.	Dynamically personalizing the downloaded Web pages according to the information about the current page and user's experience.

Table 1. Web-based databases feature.

A WBDB system is considered to be a large distributed database system and at the same time, it is different from a distributed database system in the following [4]:

- ***Number of users*** - a WBDB system should be able to support large number of transactions with reasonable response time Recovery of the lost transactions in these systems is an important task for reliable performance;
- ***Transaction processing*** - for WBDB, even a simple transaction may hold lock for a period of time that is long enough to degrade the performance of the system due to communication failure;
- ***Delivery of query results*** - should consider complex query and query with large results.

2. Architectures of Web-based databases

Architecture is a subject of design and implementation and reflects the spatial arrangement of application data and the spatial-temporal distribution of computation. There are different WBDB frameworks according to various technologies and requirements. Generally speaking, WBDB can be considered as a single huge database as well as multiple data sources. There are a lot of technologies that can be used for WBDB. Languages for web applications and web servers are Java, PHP, Perl, HTML, DHTML, XML, SQL etc. Access technologies include CGI, JavaScript, Servlet, JDBC, and ODBC. Common enterprise databases include Oracle, Sysbase, Informix, DB2, MySQL, SQL-Server.

2.1 Two-tier Architecture of WBDB

The minimal spatial configuration of a WBDB is the two-tier architecture and it closely resembles the traditional client-server paradigm. The two-tier solution clients are thin, and are lightweight applications responsible only for rendering the presentation. Application logic and data reside on the server side. Technologies involved are JDBC, XML, and SQL.

2.2 Three-tier Architecture of WBDB

The three-tier architecture contains generally client, application server and data server. A complete WBDB system requires these three essential components although they can represent various types of technologies.

In the three-tier model of a database gateway, the client component is the client API library, which consists of client-side APIs. They determine the format and meaning of the requests that the client applications may issue. The Application Server is the component that owns translation and mapping mechanisms. It transforms the client API to the DBMS server's API, and vice versa for the data returned to the clients. The server API library on the database server-side is the Data Server component. It manages the database service available to the clients. The services change in terms of authentication from the DBMS.

The transaction-processing monitor model is also a three-tier architecture. In this context, client application consists of the user interface functions, such as screen logic, screen handling, input handling, and some validation functions. Application server provides all of the details of application services. Re-

source managers can provide all of the lower-level services, such as communication between the database and the application services.

The extended client/server model is a typical three-tier architecture. In this model, the client Web browser sends requests to the Web server. The Web server transfers the requests to a database server. After the database server processes the requests, the results are retrieved to the client Web browser by the reverse pathway. In the transition, the web server can handle the results from the database.

In the multi-distributed databases (MDBS) scenario, the Web server requests the MDBS to retrieve the required data. The server does this by issuing a global-level SQL query to the MDBS. The MDBS then decomposes the whole query and generates the local queries according to various features of engaging database servers. Then these local queries can be issued to corresponding database servers that may be managed by the DBMS servers. But these DBMS servers can be accessed through all sorts of database access technologies. The MDBS integrates the local results it receives from all the database servers and finally presents a global result to the web server. In this case, the MDBS handles all the operations including data locating, interrelating, and integrating. The web server just sends the requests from clients, which is different from the typical client/server model.

All the technologies can be used in the three-tier architecture according to different user requirements. The three-tier or even n-tier models are essential models to structure a WBDB.

2.3 Hybrid Architecture of WBDB

There are several ways of combining various technologies into the Web or a database to enhance the performance of WBDB. A general architecture is to apply agent-based computing concepts in building WBDBs, witch is also a three-tier architecture (the Application Server layer contains the Web Server component and Agents).

In an agent-based scenario, a client sends either data or data and programs over the Web server that activates the agent. The agent then processes the requested data using its own programs or using the received programs. After the completion of the preliminary processing, the agent will send the data/program/medium result to the application server for further processing. Then the Web server communicates with the database, and the database server finishes the manipulation to the database and transfers the results to the Web server. The Web server will return the results back to the client directly or via the agent.

3. Web Based Database Access Technologies

Building WBDBs involves many technologies, such as database, Web server, Web browser, application server, SQL, CGI, JAVA and so on. Some technologies work for the interface; others may deal with the database access, or merge everything together, or just applications. The existing technologies used in WBDB can be classified as technologies [4]:

- Traditional Web (generation 1) - HTML, HTTP, and CGI;
- Faster and more interactive Web (generation 2) - JavaScript, Server-side API;
- Java-based Web (generation 3) - Java, JDBC, Servlet;
- XML, Client/Mobile Agents/Server (a new generation).

Due to the enormous market for WBDBs, there are all sorts of technologies developed to satisfy the actual requirements. CORBA, RMI, DCOM/.Net Remoting represents the most important middleware for WBDB.

JavaScript, Server-side API WBDB

Despite CGI-based framework advantages (simplicity, language independence, Web server independence, wide acceptance), the CGI approach has some problems (new processes for each CGI script, the database server logon and logout for every query submitted, resulting in serious resource waste and make the communication between clients and servers very slow).

Netscape's Server API (NSAPI), Microsoft's Internet Server API (ISAPI) and JavaScript are alternatives to CGI. We can view genera-

tion 2 as Server-side APIs offer much less resource-intensive access to external services. Replacing CGI with server-side JavaScript and adding client-side JavaScript significantly alters the application design domain.
JavaScript is a scripting language embedded in an HTML page. It can respond to user events such as mouse clicks, form input, page navigation, and validation and alerts. Client-side JavaScript can popup windows. In the windows, some functions, such as calculations, swapping image and controlling the GUI components can be executed. Database searching can also be simulated in the client-side JavaScript. JavaScript that runs in the Web Server processes some functions of the Web Browser. Unlike client-side JavaScript, server-side JavaScript has access to host resources, external programs and databases. The client and server-side JavaScript framework of WBDB is shown in Figure 1 [5].

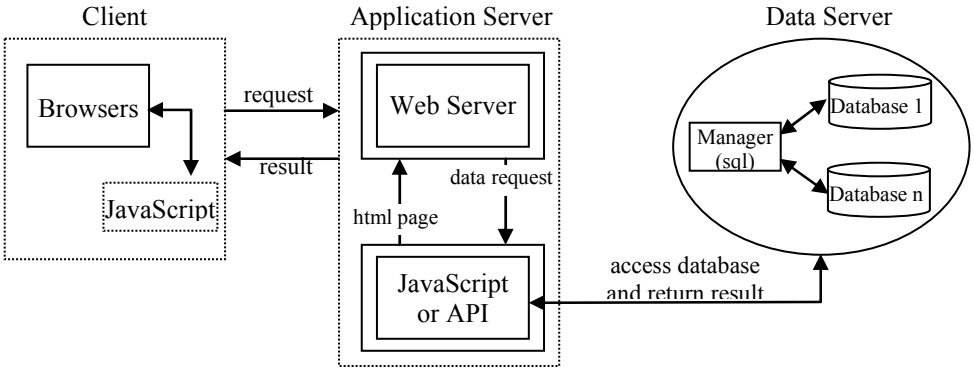


Figure 1. Client and Server-side JavaScript of WBDB

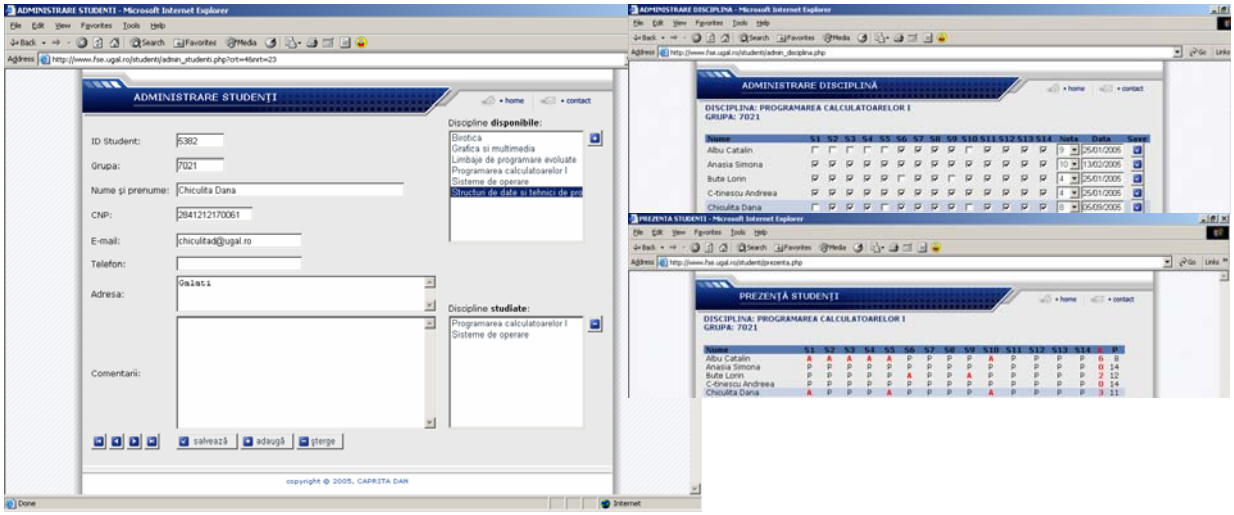
Compared with the CGI-based framework, the JavaScript/API approach works more efficiently. The transportation speed is fast. Its shorter server-side response time allows for designs of novel Web user interface that programmers would never even consider with CGI. The same language on client and server made communication and programming easier.
There are also some shortcomings for Generation 2. Due to the limitation of GUI features, user interface of Generation 2 is still HTML. It must run inside a browser. It requires the use of frames in order to maintain a persistent visual user interface context while updating another GUI control using values from a database. It cannot send the user interface an unsolicited message from the server.

4. Case Study

We have developed a generation 2 WBDB management systems for students activity. It uses PHP server-side scripts and JavaScript client-side scripts. It runs on a Windows Apache Web server that has PHP services installed and MySQL database server. We have chosen PHP because it is available for many different operating systems (free Unix like operating systems such as Linux and FreeBSD, commercial Unix versions such as Solaris, or on different versions of Microsoft Windows) and has many strengths compared to some of the main competitors (Perl, Microsoft Active Server Pages, Java Server Pages, and Allaire Cold Fusion):

- High performance;
- Portability;
- Low cost;
- Interfaces to many different database systems;
- Built-in libraries for many common Web tasks;
- Ease of learning and use;
- Availability of source code

MySQL is similarly versatile.



PHP code snippet of admin_studenti.php:

```
<?php require_once('bd.php');
session_start();
if($_SESSION['logat_nivel_acces']<=1)
  header("Location:index.php");

mysql_select_db($database_bd, $bd);
$editFormAction = $_SERVER['PHP_SELF'];
if (isset($_SERVER['QUERY_STRING']))
 {$editFormAction .= "?" . htmlentities($_SERVER['QUERY_STRING']);}

//operatie salveaza
if (isset($_POST["operatie"])&&($_POST["operatie"]=='salveaza'))
 {$updateSQL = sprintf("UPDATE studenti SET nume=%s, adresa=%s, grupa=%s,
  cnp=%s, cometarii=%s, telefon=%s WHERE id_student=%s",
$_POST['numetextfield'],$_POST['adresatextarea'],$_POST['grupatextfiled'],
$_POST['cnptextfield'], "text"),$_POST['comentariitextarea',
$_POST['telefontextfield'],$_POST['id_studenttextfield']);
  mysql_query($updateSQL, $bd) or die(mysql_error());
 }
```

Security was implemented using password access authentication (PHP md5 function), Apache's basic authentication (.htaccess), and secure connections to the MySQL database server (mysql_connect).

JavaScript snippet of admin_studenti.php:

```
<script language="javascript">
function adauga_disciplina_studiata()
{var frm=document.all["frm_student"];
 if(frm.discipline_disponibile.selectedIndex==-1)
     {alert("Selectaþi o disciplinã!");return;}
 var val_element=frm.discipline_disponibile.options
                 [frm.discipline_disponibile.selectedIndex].value;
 //verific daca mai exista disciplina
 for (var i=0;i<frm.discipline_studiate.options.length;i++)
   if (frm.discipline_studiate.options[i].value= =val_element)
      {alert("Studentul este înscris\nla această disciplinã!");return;}
 var element=document.createElement("OPTION");
 element.text=val_element;element.value=val_element;
 frm.discipline_studiate.options.add(element);
}
</script>
```

Conclusions

With the growing popularity of the Internet and the Web, there is a fast growing demand for access to database management systems from the Web. World Wide Web is a gigantic database with enormous potential applications in business, science, engineering, and education. As a consequence, the number of potential users of WBDBs is very large. A complete WBDB consists of user interfaces generally displayed on the Web browser, Web application server, and database server responsible for data manipulation.

A Web-based distributed database is a key component of many Internet related applica-

tions. The WBDB approach allows data to be represented in objects (consisting of data and methods that manipulate the data) and the access of these objects is open to anyone with the correct access rights. Information stored in a WBDB is independent of any particular software. Access to the WBDB can be easily integrated into any user interface, such as a conventional WWW browser or a particular application program. WBDBs have a great potential in electronic commerce, information retrieval and multimedia applications.

References

[1] T. Converse, J. Park, C. Morgan - *PHP5 and MySQL Bible*, Wiley Publishing, 2004

[2] G. Coulouris, J. Dollimore, T. Kindberg - *Distributed Systems. Concepts and Design*, Addison-Wesley, 2001

[3] S. Hariri, M. Parashar - *Tools and Environments for Parallel and Distributed Computing*, John Wiley & Sons, 2004

[4] W. Jia, W. Zhou - *Distributed Network Systems. From Concepts to Implementations*, Springer, 2005

[5] A.S. Tanenbaum, M. van Steen - *Distributed Systems. Principles and Paradigms*, Prentice Hall, 2002.