ABSTRACT
In recent years the World Wide Web has been elected as an ideal platform for developing Internet applications, thanks to its powerful communication paradigm based on multimodality and browsing, and to its open architectural standards that facilitate the integration of different types of content and systems.

A Data-Intensive Web Application (DIWA) is a web-enabled system for the publication and management of large data collections, typically stored in one or more database management systems.

This paper presents an overview of the different tools & techniques for developing Data-Intensive Web Application. It emphasizes on performance analysis on some of these tools based on response time for retrieving data from a backend server based database. Parameters for this analysis were chosen as number of records of the database vs. response time for retrieving information. The analysis was performed using PHP as server side scripting language running on windows machine using Apache web server.

This paper also includes a case study of Accounts Classification Chart of Ministry of Finance, Government of Bangladesh as well as the development of Online Classification Chart as a data-intensive web application. The application is built on PHP, MySQL technology which will run best on Apache HTTP server regardless of OS environment.

KEYWORDS: DIWA, Web Server, Apache, PHP, MySQL.

1. THE WEB APPLICATION FRAMEWORK
In recent years the World Wide Web has been elected as an ideal platform for developing Internet applications, thanks to its powerful communication paradigm based on multimodality and browsing, and to its open architectural standards, which facilitate the integration of different types of content and systems. Applications for the Internet in such domains as electronic commerce, digital libraries, and distance learning are characterized by an unprecedented mix of features that makes them radically different from previous applications of Information Technology (IT) [1]. Using the Web as a development environment is a relatively new phenomenon. The Web as a development environment can seem rather confusing. Because of the distributed nature of the Web, a Web application can be composed of many parts, using a variety of technologies. Figure 1 shows the various parts that make up the application framework.
2. DATA-INTENSIVE WEB APPLICATION

2.1. What is DIWA
A data-intensive Web application (DIWA) is a Web-enabled software system for the publication and management of large data collections, typically stored in one or more database management systems [2]. DIWA constitute the most diffused class of applications found on the Web today, and their industrial relevance is an established fact. Conceptually we can understand DIWA simply as the Internet extension of a client server application. However, since the Web by nature is stateless and the user interface (i.e. some kind of form) is separate and disconnected (between requests) from the application it runs on (the server), maintaining state information from request to request and user to user requires the use of special tools and capabilities. DIWA is dynamic in nature, because data can be retrieved and stored dynamically at the servers where one or more databases reside. For this reason, sometimes the data-intensive Web applications are simply called dynamic Web page or data-based Web pages [3].

2.2. Life Cycle of DIWA
Although there is still no consensus on a general model of the lifecycle of a data-intensive Web application [4], a scheme of typical activities involved in constructing a data-intensive Web application can be obtained by interpolating the lifecycle models of traditional information systems. Figure 2 illustrates the life cycle model used as reference:

2.3. Architecture of DIWA
Data-intensive Web applications are generally of 3-tier architecture (Figure 3). In 3-tier architecture, an application is broken up into three separate logic layers, each with a well-defined set of interfaces. The first tier is referred to as the presentation layer and typically consists of a graphical user interface (GUI) of some kind. The middle tier consists of the application logic and the third tier – the data layer – contains the data that is needed for the application.

3. TOOLS & TECHNIQUES FOR DIWA

3.1. HTML
HTML Stands for Hyper Text Markup Language and is the authoring language used in the creation of documents for the World Wide Web. HTML was initially created for use as a universal common document language for the World Wide Web. XHTML (eXtensible HTML) is going to replace HTML.

3.2. Client-side Scripting
3.2.1. JavaScript
A scripting language developed by Netscape to enable Web authors to design interactive sites. Although it shares many of the features and structures of the full Java language, it was developed independently. JavaScript can interact with HTML source code, enabling Web authors to spice up their sites with dynamic content.

3.2.2. VB Script
A scripting language developed by Microsoft. Microsoft Visual Basic Scripting brings active scripting to a wide variety of environments, including web client scripting in Microsoft Internet Explorer and Web server scripting in Microsoft Internet Information Server.

3.3. Server-side Scripting
3.3.1. CGI
A set of rules that describe how a Web Server communicates with another piece of software on the same machine, and how the other piece of software, the 'CGI program' talks to the web server. Any piece of software can be a CGI program if it handles input and output according to the CGI standard.

3.3.2. PERL
Practical Extraction and Report Language (PERL) is an interpreted language used for the development of CGI scripts. PERL provides easy means for Web developers to process text strings provided by the Web server according
to the CGI standard. The vast majority of scripted programs on Websites running on the UNIX operating system are written in PERL.

### 3.3.3. ASP

Active Server Pages enable web developers to make their sites dynamic with database driven contents. The code is mainly written in VB Script, and it is produced on the server of the web site instead of the browser of your web site visitors. ASP.NET has replaced older ASP technology.

### 3.3.4. Cold Fusion

A Cold Fusion application is very simply a collection of pages, similar to a static Web site. But unlike the pages in a static Web site, the pages in a Cold Fusion application include the server-side Cold Fusion Markup Language (CFML) in addition to HTML.

### 3.3.5. Java Servlet

Java Servlet is an applet that runs on a server. The term usually refers to a Java applet that runs within a Web server environment. This is analogous to a Java applet that runs within a Web browser environment. Java Servlets are becoming increasingly popular as an alternative to CGI programs [5].

### 3.3.6. JSP

A server-side technology, Java Server pages (JSP) are an extension to the Java Servlet technology that was developed by Sun. JSP have dynamic scripting capability that works in tandem with HTML code, separating the page logic from the static elements – the actual design and display of the page. Embedded in the HTML page, the Java source code and its extensions help make the HTML more functional, being used in dynamic database queries, for example. JSP are not restricted to any specific platform or server. JSP is also known as Scriplet [5].

### 3.3.7. PHP

PHP (recursive acronym for "PHP: Hypertext Preprocessor") is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML pages. PHP is mainly focused on server-side scripting, so one can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. But PHP can do much more.

### 3.4. Data Base Connectivity

#### 3.4.1. ODBC

Open Database Connectivity (ODBC) is a standard database access method developed by Microsoft Corporation. The goal of ODBC is to make it possible to access any data from any application, regardless of which database management system (DBMS) is handling the data. ODBC is gradually being replaced with OLEDB.

#### 3.4.2. JDBC

JDBC is a Java API that enables Java programs to execute SQL statements. This allows Java programs to interact with any SQL-compliant database. Since nearly all relational database management systems (DBMS) support SQL, and because Java itself runs on most platforms, JDBC makes it possible to write a single database application that can run on different platforms and interact with different DBMS.

### 3.5. Java Applets

Within the browser environment, Java programs are called applets. The applet can be downloaded onto client computer. The applet comes to the browser in byte code. If a browser supports Java, it interprets this byte code and executes it on the client machine.

### 3.6. XML

Extensible Markup Language (XML) is designed to enable the use of SGML on the World Wide Web. XML is a meta-language (a way to define tag sets) that allows you to design your own customized markup language for many classes of documents. XML is intended to deliver information, not just pages.

### 4. THE PERFORMANCE ANALYSIS

#### 4.1. The Approach

As part of our project, we did some comparison analysis. Our objective was to find the performance of different techniques based on response time. Due to shortage of time, we only concentrate on PHP as server side scripting among the different techniques present these days, such as ASP, ASP.NET, Cold Fusion, JSP, Java Servlet, Perl, PHP etc. And for backend database, we only consider MySQL and MS Access among the different DBMS, such as Oracle, SQL Server, PostgreSQL, Informix, MySQL, MS Access. We created a dummy flat database consisting of only one table. Number of records in the database varies from one thousand to one million. The query we used was to find all the records from the database. We wrote three different scripts to retrieve the data.

- To connect to the MySQL database using PHP’s built in MySQL functions.
- To connect to the MySQL database using PHP’s ODBC functions, this required the MyODBC driver in the system.
- To connect to the MS Access database using PHP’s ODBC functions.

In all the above scripts, we use PHP’s built in microtime() function and a user defined getmicrotime() function to calculate the response time of the query. We run these scripts under Apache HTTP server configuring it to use
PHP first as a CGI then as SAPI module under Windows. To get a better accurate value, we run the query scripts five times for the number of records one thousand to one million.

The hardware configuration of the system we used for this experiment is given below:

- Operating System – Microsoft Windows XP Professional 5.1.2600
- Processor – AMD Athlon K7 MMX, 3DNow 700 MHz
- Memory – 256 MB SD RAM
- AGP Card – Asus V7100 GeForce2 MX 64 MB SD RAM

4.2. Findings of the Analysis

When we found that there is a very little difference between CGI and SAPI module, we chose to work further only with SAPI module. The results we got from the experiment are given below:

Table 1: Response Time (in second) for various numbers of records

<table>
<thead>
<tr>
<th>Records</th>
<th>MySQL (built in func.)</th>
<th>MySQL (using ODBC)</th>
<th>MS Access (using ODBC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 thousand</td>
<td>0.031192</td>
<td>0.050797</td>
<td>0.047061</td>
</tr>
<tr>
<td>25 thousands</td>
<td>0.543405</td>
<td>0.753588</td>
<td>0.555350</td>
</tr>
<tr>
<td>50 thousands</td>
<td>1.093525</td>
<td>1.493461</td>
<td>1.111815</td>
</tr>
<tr>
<td>75 thousands</td>
<td>1.577178</td>
<td>2.267985</td>
<td>1.624719</td>
</tr>
<tr>
<td>100 thousands</td>
<td>2.069578</td>
<td>2.842884</td>
<td>2.138066</td>
</tr>
<tr>
<td>250 thousands</td>
<td>4.958957</td>
<td>7.209894</td>
<td>5.239326</td>
</tr>
<tr>
<td>500 thousands</td>
<td>10.595971</td>
<td>14.153220</td>
<td>10.596044</td>
</tr>
<tr>
<td>750 thousands</td>
<td>15.520803</td>
<td>21.767626</td>
<td>21.250313</td>
</tr>
<tr>
<td>1 million</td>
<td>20.904658</td>
<td>29.266117</td>
<td>29.825840</td>
</tr>
</tbody>
</table>

We plot two comparison graphs based on the above data:

- One graph for the comparison between MySQL and MS Access both using ODBC.
- Another graph for the comparison between MySQL built-in and MS Access using ODBC.

In the comparison between MySQL and MS Access both using ODBC, we see that (Figure 4) for number of records up to 100 thousands there is a negligible difference between them with MS Access having a slight better response time. But then up to 750 thousands records MS Access response time is very good compared to MySQL. And then after 750 thousands records both seem to have same response time with an indication of MS Access response time getting higher.

Meanwhile, in the comparison between MySQL built-in function and MS Access, the scenario is quite different. MySQL seems to have a better response time always over MS Access (Figure 5). For number of records up to 500 thousands, the difference is very little. But then suddenly the response time for MS Access is very higher than the response time for MySQL. For 1 million records the difference is around 9 seconds.

From this, we come to the conclusion that for small databases there is no big difference in response time between MySQL and MS Access. But as databases grow larger, MySQL out performs in response time comparing to the response time of MS Access.

5. ONLINE CLASSIFICATION CHART

5.1. What is Classification Chart

Classification Chart is a collection of codes with description to store all the budget receipt and payment information for the government of People’s Republic of Bangladesh. The Ministry of Finance and the Comptroller and Auditor General of Bangladesh use this chart. A full over-
view and all the definitions regarding the Classification Chart can be found in the Classification Chart book [6].

5.2. The Online version
At this moment the Classification Chart is just a book containing all the information. As part of project, we tried to make the Classification Chart online.

5.3. The Scope
The Online Classification Chart will help user to search information regarding codes and their descriptions. User can search information within all the four levels of code. The tool is divided into four specific modules to help the user to use it more easily. The modules are listed below:

- Module 1: Search by Code
- Module 2: Search by Description
- Module 3: Search a Range
- Module 4: Help & Glossary

All the features of the Classification Chart are maintained in this online version. We follow the same structure of Classification Chart to build this tool. The tool is still on development process. A lot of improvements can be added to this tool. But the tool present now is good enough for searching the codes and their descriptions. We hope this tool will gradually change the way of searching codes and description in classification chart. [The complete report on the online version can be found by contacting one of the authors.]

5.4. Architectural Design
Among the four modules, only module 4: help & glossary need not to communicate with the backend database. The other three modules connect to the database, request a query, and output the response to user. The architectural design of the application is given in Figure 6. Some abbreviation has been used on the diagram: M1 – Module 1, M2 – Module 2, M3 – Module 3, M4 – Module 4, S – Apache Server, DB – MySQL Database, P – Process.

5.5. Detail Design

Figure 7: Component Diagram of Online Classification Chart

Figure 8: Database Diagram of Online Classification Chart

6. LIMITATIONS
Due to short of time and resource, there were some limitations in both part of our project work:

1. We only choose MySQL and MS Access among so many databases.
2. We only consider the number of records as the parameter.
3. We construct only Flat database.
4. We test it only in Microsoft Windows XP.
5. For case study we didn’t consider the ANNEX of the classification charts.

7. CONCLUSION
Our Project work was divided into two parts. After evaluation of response time and working on the case study we have come to the following conclusions:

1. For number of records up to 500 thousands both MySQL and MS Access, the response time is almost same with MySQL slightly better.
2. For number of records over 500 thousands MySQL response time is far better than MS Access response time.
3. MySQL performs better in response time when PHP connects with built in MySQL methods, comparison to connecting with PHP ODBC methods.
4. For response time analysis, both PHP as a CGI and PHP as ISAPI module perform almost same.
5. In our case study of Classification Chart, there are some data redundancies that are removed in the online version.
6. Online version of the classification chart reduces the search time significantly.

8. SCOPE OF FUTURE WORKS
The followings are some of the scope of future works:
1. Testing response time analysis among other front end techniques such as ASP, JSP, Cold Fusion, etc.
2. Testing for more database applications such as Oracle, MySQL, PostgreSQL, etc.
3. Testing for both flat and relational databases.
4. Increasing testing parameters.
5. Test on different OS and Web server.
6. More enhancement of the Online Classification Chart

REFERENCE:
[22]. http://www.mysql.com
[23]. http://www.microsoft.com