

CS3743 Introduction to Database Systems Internet Databases

Weining Zhang

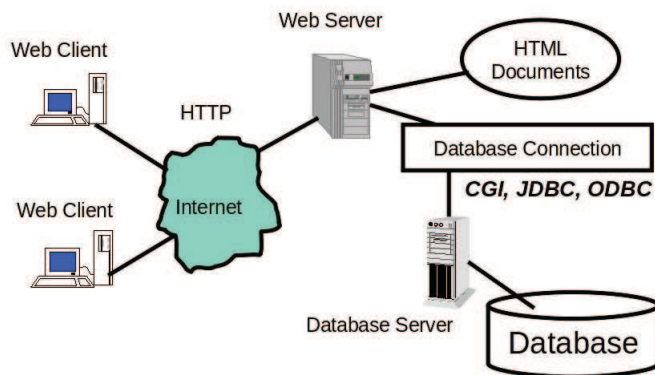
Department of Computer Science
University of Texas at San Antonio

August 4, 2009

Outline

- 1 Overview
 - Internet and Databases
- 2 World Wide Web
 - HTML
 - HTTP
- 3 Web Access to Databases
 - Common Gateway Interface
 - Application Servers

Web-Based Database Application

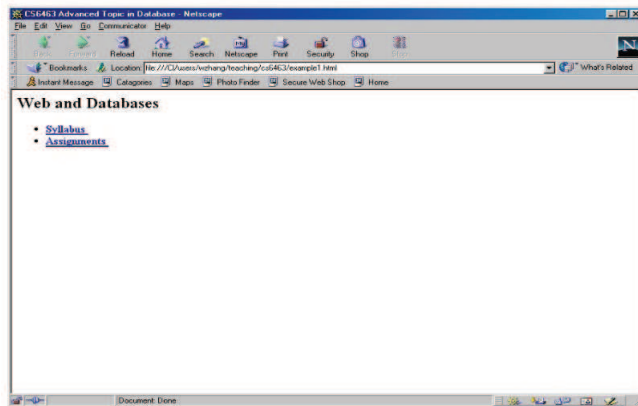


Web-Based Database Application

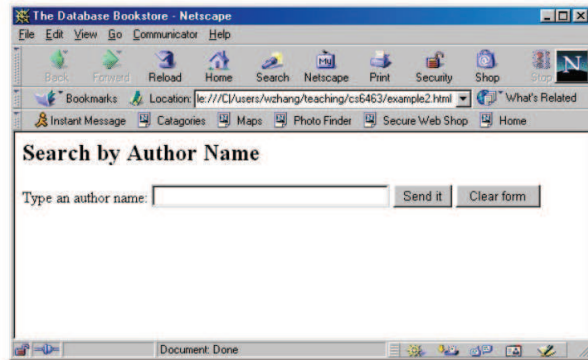
- User interface is provided by web browser
- Application is hosted on a web server or an application server
- Access data in a database server through programming interfaces
- There are many ways to implement web-based database applications

- The **HyperText Markup Language** (HTML) is a language to mark up documents using tags
- HTML documents are stored on or generated by a web server and are displayed by a web browser or consumed by other types of web client
- The **HyperText Transmission Protocol** (HTTP) is a standard protocol for web server and web client to communicate with each other

```
<html>
<title> CS 6463 Advanced Topics in DB </title>
<body>
<h2> Web and Databases </h2><hr>
<ul>
  <li> <a href="file:/home/cs6463/syllabus.html">
    <b> Syllabus </b> </a> </li>
  <li> <a href="assignment.html">
    <b> Assignments </b> </a> </li>
</ul>
</body>
</html>
```



```
<html> <head>
<title> The Database Bookstore </title>
</head> <body>
<h2> Search by Author Name </h2>
<form action="cgi_bin/find_books.cgi" method=post>
  Type an author name:
  <input type="text" name="authorName" size=30 maxlength=50>
  <input type="submit" value="Send it">
  <input type="reset" value="Clear form">
</form>
</body> </html>
```



- Have 9 different input types, including
 - Text: a single-line text entry field.
 - Checkbox: a box to be set/reset.
 - Object: a JavaBean object.
 - Submit: a button to submit data.
- Can specify variables (with name=varName) and default values (with value=defaultValue)

- HTTP defines a single stateless transaction per connection.
 - The client makes a connection with a server.
 - The client makes a request by sending the server a message with a header and an optional body.
 - The server responds by sending a message with a header and an optional body.
 - The server closes the connection.
- There are several types of requests and responses.

The Uniform Resource Locator (URL) is used in HTTP to uniquely identify a resource (such as a web page)

Example

The URL

```
http://www.cs.utsa.edu/~wzhang/cs3743
```

Indicates that

- communication protocol is http
- web server address is www.cs.utsa.edu
- web page on the server is at /~wzhang/cs3743/index.html

HTTP

A Sample Request

```
GET /path/file.html HTTP/1.0
Accept: text/html
Accept: image/jpeg
If-Modified-Since: Wed, 25 Sep 1996 17:23:31 GMT
Referer: http://www.utoronto.ca/webdocs/webinfo.html
User-Agent: Mozilla/1.01
```

HTTP

A Sample Response

```
HTTP/1.0 200 Document follow
Date: Sat, 13 Dec 1997 16:04:09 GMT
Server: NCSA/1.5.2
Content-type: text/html
Last-modified: Thu, 03 Oct 1996 16:03:27 GMT
Content-length: 139

<html><head> <title> Test HTML file </title> </head>
<body> <h1> This is a test file</h1>
<p> So what did you expect, art?
</body></html>
```

HTTP

Header-Only Communication

Client:

```
HEAD /path/file.html HTTP/1.0
User-Agent: HEAD TEST PROGRAM
From: name@domain.name.edu
```

Server:

```
HTTP/1.0 200 OK
Date: Sat, 13 Dec 1997 16:04:09 GMT
Server: Apache/1.1.1
MIME-version: 1.0
Content-type: text/html
Content-length: 139
Last-modified: Thu, 03 Oct 1996 16:03:27 GMT
```

HTTP

Send a Form Using HTTP Get Method

```
GET /cgi-bin/form?srch=dogfish&srch_type=Exact+Match&svr=Canada&svr=Sweden HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/3.02 (Win95; 1)
Host: smaug.java.utoronto.ca:2021
Accept: image/gif, image/jpeg, */*
Accept-Language: en-US, fr
```

Send a Form Using HTTP Post Method

```
POST /cgi-bin/form1 HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/3.02 (Win95; 1)
Host: smaug.java.utoronto.ca:2021
Accept: image/gif, image/jpeg, */*
Accept-Language: en-US, fr
Content-Type: application/x-www-form-urlencoded
Content-Length: 58

srch=dogfish&srch_type=Exact+Match&srvr=Canada&srvr=Sweden
```

Other Features of HTTP

- Other methods: PUT, DELETE, etc.
- Many more header fields.
- Lots of status codes defined.
- Can limit access to a specific internet domain.
- Basic & digest (encrypted) user authentication.
- Data encryption (secure HTTP, https).
- Proxy & server caching.

Framework of CGI

- A gateway program can be any program in a special directory on a web server
- Client calls a gateway program by sending data to the web server with the URL of the program
- The web server passes data from the client to the gateway program via a CGI process
- The gateway program processes the data, and optionally connects to a database server via a special database API (application program interface)
- The gateway program returns results to the web server via CGI
- The web server forwards the result back to the web client

An CGI Example

- Via a form, a user searches a database for books by a book title.
- A Sybase database has a relation

Books(Author, Title, Publisher, Year, Pages)

- A gateway perl program will access the Sybase database to retrieve books of the given title. If a book is found, the gateway program returns an HTML table containing all matched books.

An CGI Example

```

<html>
<head>
<title> CGI Publishing Co. </title>
</head> <body>
<h1> Book Search </h1> <hr>
<form action="/cgi-bin/books.pl" method="get">
<em> Please enter the title of the book that you would like to search
for </em>
  <p> <input type="text" name="Book" size=40>
  <p> <input type="submit" value="Search">
  <input type="reset" value="Clear Title">
</form>
<hr>
</body>
</html>

```

An CGI Example

```

#!/usr/local/bin/syberpl
require "syberpl.pl"; # Perl library for Sybase
$user = "scott";
$password = "mkh8hd9v2ek";
$server = $ENV{'DSQUERY'} || "Books";
$dbproc = &dblogin($user, $password, $server); # connect to database
@fields = ('Author', 'Title', 'Publisher', 'Year', 'Pages');
$tablehead = "CGI Publishing Co Book DB";
&parse_form_data(*DB); # get input data
($book_name = $DB{'Book'}) =~
  s/^\\s*(.*)\\b\\s*$/$1/; #remove spaces
if ($book_name =~ /^[\\w\\s]+$/) {
  &dbcmd($dbproc, "select * from Books
    where Title = '$book_name'");
  &dbsqlxexec($dbproc); # run SQL query
  $status = &dbresults($dbproc);

```

An CGI Example

```

if ($status == $SUCCEED) {
  while((@books=&dbnextrow($dbproc)) {
    $book_string = join("\\0", @books);
    push(@all_books, $book_string);
  } # get the query result
  &dbexit($dbproc); # disconnect database
  &display_table($tablehead, *fields,
    *all_books, "\\0"); # create html table
} else {
  &return_error(500, "Sybase Error", "Book title not found.");
} else {
  &return_error(500, "Sybase Error",
    "Invalid character in book title.");
} exit(0);

```

An CGI Example

```

sub display_table {
  local ($tablehead, *columns, *selected_entries, $delimiter) = @_;
  local ($name, $entry);
  print "Content-type: text/html", "\\n\\n";
  print "<html>", "\\n"; # create html table
  print "<head><title> $tablehead ", "\\n";
  print "</title> </head>", "\\n", "<body>", "\\n";
  print "<table border=2>", "\\n", "<caption>", $tablehead;
  print "</caption>", "\\n", "<tr>", "\\n";
  foreach $name (@columns) { print "<th>", $name, "\\n"; }
  foreach $entry (@selected_entries) {
    $entry =~ s/$delimiter/<td>/go;
    print "<tr>", "<td>", $entry, "\\n";
  }
  print "</table>", "\\n", "</body></html>", "\\n";
}

```

Pros and Cons of CGI

- CGI is flexible. Extending HTTP server functionality without changing the server.
- CGI is slow. It is expensive for OS to start a new process.
- CGI may not scale well. Large number of requests for gateway programs will require the system to start many parallel processes.
- Server API may improve speed, but APIs for Netscape, Apache, etc. are not compatible.

Other Server-Side Processing

- Java Servlets: Java programs that run on the server and interact with the server through a well-defined API.
- JavaBeans: Reusable software components written in Java.
- Java Server Pages and Active Server Pages: Code inside a web page that is interpreted by the web server

Application Server

- In CGI, each page request results in the creation of a new process: very inefficient
- Application server: Piece of software between the web server and the applications
- Functionality:
 - Hold a set of pre-forked threads or processes for performance
 - Database connection pooling (reuse a set of existing connections)
 - Integration of heterogeneous data sources
 - Transaction management involving several data sources
 - Session management