# Server-side Processing

Web server

- a server-side application program that runs on a host computer and responds to requests for web pages (i.e. services HTTP requests)
- Other possible functions:
  - access control
  - run scripts/programs
  - management and administration of server functions and Web site contents
  - Log transactions
- Two main commercial web servers:
  - Apache (64.6%)
  - Microsoft Internet Information Server (IIS) (23.5%)
  - The Netcraft Web Server Survey, October 2003, retrieved from http://www.netcraft.com/survey/, Oct. 10, 2003

Market Share for Top Servers Across All Domains August 1995 - October 2003
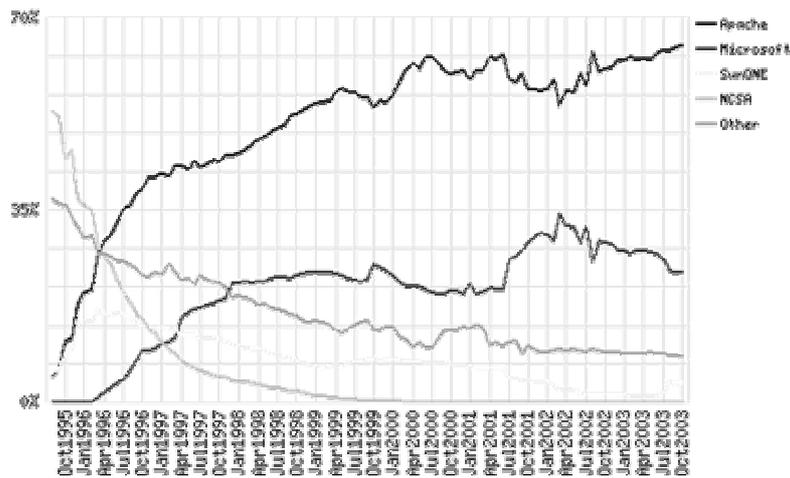
3

Static vs. Dynamic Web pages

- Most basic function of web servers is to distribute HTML documents
- HTML = Static
  - all clients see the same content

- However, eCommerce is more than just displaying a "static catalog"
  - we want to customize the web page to suit the information the user is looking for?
  - We want to send real-time data (i.e. current date/time)?
  - We want to do sophisticated processing on user-supplied data and report results

4

Client-side vs. Server-side processing

- Dynamically create content that doesn't reside in the HTML pages
  - Create and modify information "on-the-fly"
  - Data is often part of databases that resides elsewhere (on other disks or even other servers)

5

Client- vs. Server-side Processing

- Client
  - Validation, interactivity, enhancing web pages
  - Reduces traffic to/from server
  - Reduces the amount of the work the server needs to perform (run 100 programs on 100 client computers instead of 100 program on a server)
  - Dependent on browser functionality (e.g. is the specific technology supported by the individual browser)
  - Scripts are visible
  - Great way to distribute little application programs (e.g. currency converter)
- Server
  - Generate custom responses for clients
  - Runs exclusively on the server – no cross-platform issues
  - Scripts not visible to client – more proprietary

6

Server-side processing

🔳 Need an intermediatary to take the inquiries from the client and gather the information from the database

      → "middleware"
      → "three-tier-architecture"

CGI Programming (Common Gateway Interface)

🔳 "Set of standard methods and routines used to write stand-alone software programs that know how to receive requests from a Web server and return data to the server." [1]

🔳 "a specification for transferring information between a Web server and a CGI program."[2]

🔳 Allows you to[3]:
- create an executable that the web server may call on demand
- pass incoming HTTP GET or POST data to the CGI script
- send CGI generated answers back to the browser

🔳 CGI programs can be written in a variety of programming languages: Perl, VB, C/C++, Java, etc.

Sources: 1) Turban, Web Programming, Appendix C. 2) Webopedia. Retrieved Sept. 22, 2002 from: http://www.webopedia.com/TERM/D/DOM.html,. 3)CGI (Common Gateway Interface), Selena Sol, October 4, 1999, retrieved Sept. 22, 2002 from: http://www.wdvl.com/Authoring/Tools/Tutorial/cgi.html,.
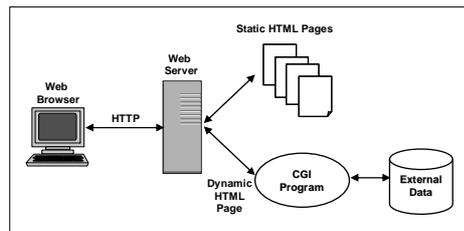
Perl

- *"Practical Extraction and Report Language"*
- freely available for most operating systems
- powerful string manipulation functions
- easy (?) to learn; can be used to implement most sophisticated processing
- "monolithic code"

- Good (quick) introduction: http://www.cgi101.com/
- presentation by Richard Carpenter

CGI Programming (Common Gateway Interface)


```
<HTML>
<HEAD>
<TITLE>Simple CGI Example</TITLE>
</HEAD>

<BODY BGCOLOR = "#FFFF80">

<A href="http://www.cs.dal.ca/~tt/cgi-bin/hello.cgi"> Hello World
   CGI Script</A>

</BODY>
</HTML>
```

hello.cgi

```perl
#!/opt/bin/perl                    ← Where the perl interpreter is located
#hello.cgi: a simple Perl CGI example
                                   ← What type of data is being sent to the browser
print "Content-type: text/html\n\n";  ← Two carriage returns
#Note that we output two carriage returns after the
#content type. This is very important as it marks
#the end of the CGI "header" and the beginning of
#the document to be sent to the browser.
                                   ← Generate the HTML head & body
print "<head>\n";
print "<title>Hello World CGI</title>\n";
print "</head>\n";
print "<body>\n";
print "<h1>Hello, World !</h1>\n";
print "</body>\n";
```

http://www.cs.dal.ca/~tt/cgi-bin/hello.cgi

11

PERL << PRINT Command

▪ Doesn't require a print statement on every line

```perl
print "Content-type: text/html\n\n";

print << "ending_print_tag";
<head>
<title>Hello, World</title>
</head>
<body>
<h1>Hello, World!</h1>
</body>
ending_print_tag
```

12

6

Passing Data

- Normally use forms to pass user input data to a web server

```
<html>
<head><title>Perl Example</title></head>
<body>

<p>Please enter your name:</p>
<form method="GET" action="http://www.cs.dal.ca/~tt/cgi-bin/getenv.cgi">
  <input type="text" name="Name" size="20">
  <input type="submit" value="Submit" name="B1">
</form>

</body>
</html>
```

http://www.cs.dal.ca/~tt/ECMM6010/PerlExample.htm

13

GET vs. POST

- Two methods that data can be sent to the CGI program:
  - GET
    - GET sends data appended to the URL string
      ```
      http://www.domain.com/dir/file?data1&data2&data3
      ```
  - POST
    - POST sends data after all the request headers have been sent to the server
    - Uses the "content length" header to let the CGI program know how much data to read

14

Processing Input Data

```perl
#!/opt/bin/perl
#hello.cgi: a simple Perl CGI example

print "Content-type: text/html\n\n";

#read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
#@pairs = split(/&/, $buffer);
@pairs = split(/&/, $ENV{'QUERY_STRING'});
foreach $pair (@pairs) {
    ($name, $value) = split(/=/, $pair);
        $value =~ tr/+/ /;
            $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
                $FORM{$name} = $value;
                }


print << "ending_print_tag";
<title>Hello $FORM{'name'}</title>
</head>
<body>
<h1>Hello $FORM{'name'}</h1>
</body>
ending_print_tag
```

Used for POST

Used for GET

15

Most common CGI Bugs

- Make sure your file and directory are executable
  - Give both your file and directories read and execute permissions (`chmod 755 file.cgi`)
    - Perl scripts will never work if the permissions on the file are 777
  - Make sure perl is in the directory you specified (`/opt/bin/perl`)
  - Check that your program executes ok (type it in at the command line `hello2.cgi`)
  - Have the interpreter check the syntax without actually running it (`perl -cw hello2.cgi`)
  - Make sure there are no hidden characters when transferring files from PC (`e.g. ^M`)

16

Servlets

- CGI programs in Java
- Servlets are Java programs that extend `HttpServlets` and implementing specific methods that are called by a `get` or `post` request

- Discussed in Darrel Ince, Developing Distributed and E-Commerce Applications, Chapter 7

17

Limitations of CGI Programming

- CGI programs are separate executables or interpreted scripts

- Other limitations:
  - CGI introduces security holes
  - Limited by HTTP
  - CGI can be ugly

http://wdvl.internet.com/Authoring/Scripting/WebWare/Server/CGIsux.html

18

Solution

■ Embed the processing into the web server (and web page) itself

■ Other server-side scripting languages:
  – Server Side Includes (SSI)
  – Microsoft's Active Server Pages (ASP)
  – Apache's Hypertext Preprocessor (PHP)
  – Java Server Pages (JSP)
  – ColdFusion's Markup Language (CFML)

19

Server-side scripts

■ A Web page that contains script statements interspersed with HTML tags
  – <% ….. %> specialized script tag

■ When the web page is processed it is first handed off to the appropriate server-side script processor
  – .shtml, .asp, .php, .cfm
  – The script code gets executed and the resulting page sent back to the browser via Web server

20

SSI (Server Side Includes)

- Commands are integrated into an HTML page using special tags
- The web server understands the special tags and translates the code on the fly
- The result is embedded into the HTML document which is then passed to the browser
- Syntax:
  - `<!--#command arg1="value1 arg2="value2 ... -->`
- Warnings:
  - Can be costly to parse files on heavily loaded servers
  - Can be a security risk

21

hello4.shtml

```
<html>
<title>Hello World</title>
</head>
<body>
<h1>Hello World</h1>

The Current Date is:
<!--#echo var="DATE_LOCAL"-->
<br>
<!--#exec cmd="date"-->

</body>

</html>
```

22

SSI (old technology)

- What happens:
  - Server gets request for a page (.shtml)
  - Server checks the page for SSI commands
  - Server executes those commands and inserts new values into the page
  - Server send the new page to the browser

Server Pages

- Java Server Pages (Sun) .jps
- Active Server Pages (Microsoft) .asp

- Instructions for the server are included in the web page
- The server notices the extension and looks for those instructions and executes them

JavaServer Pages (JSP) & Servlets

- Java servlets – small Java programs that run on a Web server and build Web pages. They are specified in a web page and run on the web server to modify the web page before it is sent to the user.
- Java Server Pages – web page coding standard to combine static HTML with dynamically-generated HTML (created by JSP scripts, and Java)

25

---

date.jsp

```
<HTML>
<HEAD>
<TITLE>JSP Example</TITLE>
</HEAD>
 <BODY BGCOLOR="ffffcc">
  <CENTER>
  <h2> The current date is <%= new java.util.Date()%>.
  </h2>
  </CENTER>
 </BODY>
</HTML>
```

26

Active Server Pages

- Microsoft's version to dynamically create Web pages
- Utilizes VBScript or Jscript
- Has a robust set of objects for serious programming, allows developers to code custom tags, and can instantiate server side resources (COM components)

27

hello.asp

```
<%@ Language=VBScript %>

<HTML>
 <HEAD>
<TITLE>Example 1</TITLE>
</HEAD>

<BODY bgcolor=Lime aLink=DarkTurquoise>
<P>
</P>
<% Response.Write("Hello, world!") %>
</BODY>
</HTML>
```

28

Further topics

   – .NET (Rui Ye)

   – J2EE (Rachna Juneja)

   – ColdFusion (Yu Fu)

29

References

- Definitions from Webopedia:
  http://www.webopedia.com
- Turban, Web Programming, Appendix C.
- Selena Sol Tutorials:
  http://www.wdvl.com/Authoring/Tools/Tutorial/

30