Server-Side Technologies: An Introduction

Qiang Yang

Thanks to: Professors Ethan Cerami (NYC) and Francis Lau (HKU)

Web Server

- Web server
  - A program that, using the client/server model and the HTTP protocol, serves the files that form Web pages to Web users
  - Every computer on the Internet that contains a Web site must have a Web server program
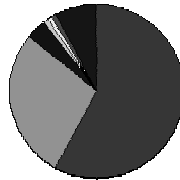

Web Server (cont'd)

- Top servers by market share (Netcraft)
  - Apache (62%)
  - Microsoft IIS (20%)
  - iPlanet (7%)
  - Netscape Enterprise (7%)

Market Share – September 2000 (285342 servers) Across ALL Domains

- Apache (57.86%)
- Microsoft (20.14%)
- Netscape 06.14%
- WebSite <1.00%
- WebSTAR <1.00%
- Zeus 09.90%
- Other 6.00%

Source: E-soft

Web Server (cont'd)

REPORT CARD
Web Servers

Feature	Weight	Netscape Enterprise Server 4.0	Microsoft Internet Information Server 4.0	Apache Software Apache Server 1.3.9
Performance	30%	4	5	3
Development	20%	3.8	4.2	4.2
Configuration	15%	5	4	4
Management	15%	4	5	4
Platform support	10%	5	2	5
Stability	10%	4	2	5
Total score		4.21	4.09	3.94
		B+	B+	B

A≥4.3, B≥3.5, C≥2.5, D≥1.5, F<1.5　A–C grades include + or – in their ranges.
Total scores and weighted scores are based on a scale of 0–5.

Source: Network Computing

Apache

- Open Source
  - Freely distributed on the Internet
  - Huge amount of end-user support
- Has earned the reputation of being the most reliable Web server available
- Uses server modules to **add functionality** to the core HTTP server
  - These can be dynamically loaded at run-time or statically compiled into the binary beforehand

---

Server Side Options

- Common Gateway Interface (CGI)
- Fast CGI
- Mod Perl
- Server Extensions
  - NSAPI
  - ISAPI
- ASP
- PHP
- Cold Fusion
- Java Servlets

Common Features

- All server side frameworks share a common set of features:
  - Read data submitted by the user
  - Generate HTML dynamically based on user input
  - Determine information about the client browser
  - Access Database systems
  - Exploit the HTTP protocol

Decision Points

- When evaluating which server side framework to use, you need to consider a number of critical factors:
  - Ease of development:
    - How easily can you build new applications?
  - Performance:
    - How fast can the framework respond to queries?
  - Scalability:
    - Can the framework scale to thousands, millions of users?
  - Security:
    - Are there any inherent security vulnerabilities?

Option 1: CGI

- Represents one of the earliest, practical methods for generating web content.
- Primarily written in the Perl programming language.
- Unfortunately, traditional CGI programs suffer from scalability and performance problems.
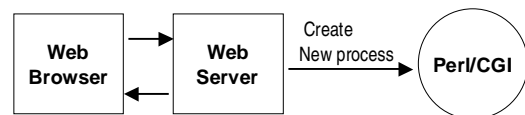- Let's examine these two problems…

CGI Architecture

1) Browser initiates request
2) Web server receives the request.
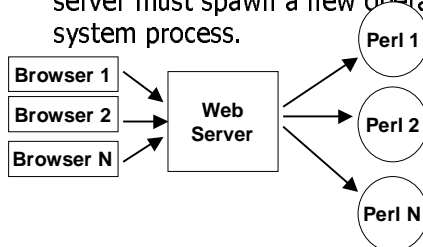3) For each request, web server spawns a new operating system process to execute the CGI/Perl Program.

CGI Architecture

- For each browser request, the web server must spawn a new operating system process.

CGI Architecture

- Spawning a new operating system process for each request takes time and memory.
- Hence, traditional CGI programs have inherent performance and scalability problems.
- Every other server architecture tries to address these problems.

Option 2: Fast CGI

- Developed by Open Market as an option for developing faster, more scalable CGI programs.
- Fast CGI works by creating a pool of processes for handling CGI requests.
- When a CGI request comes in, Fast CGI picks one of the processes from the pool and assigns it to the task.
- Without the overhead of creating new operating system processes, FastCGI is much faster than traditional CGI.
- For more information, see http://www.fastcgi.com

Option 3: Mod Perl

- A module of the Apache Web Server (now the most popular web server in the world!)
- Embeds the Perl interpreter directly within the web server.
- Perl programs are therefore precompiled.
- Because Perl is embedded within the Server, Mod Perl does not need to create a new process for each request.
- Like FastCGI, Mod Perl is much faster than traditional CGI.
- For more information, see: http://perl.apache.org

Option 4: Server Extensions

- Several web servers provide extension APIs.
  - Netscape provides NSAPI
  - Microsoft provides ISAPA
- Much like mod_perl, these programs run directly within the web server.
- Hence, server extensions are much faster than traditional CGI.
- Usually written in C/C++, and are not portable across web servers.
- For example, if you develop to Netscape NSAPI, you cannot run it on ISAPI.

Option 5: ASP

- Active Server Pages
- Runs on Microsoft's Web Server: Internet Information Server (IIS)
- Programmers add ASP code directly into their HTML pages.
- When a client requests a page, the Web Server takes the HTML page, runs the ASP code within the page, and returns a complete HTML page.
- Faster than traditional CGI, but only works on Microsoft IIS.

Option 6: Cold Fusion

- Developed by Allaire Corporation.
- Provides excellent database access and database tools.
- Great platform for rapid prototyping and rapid development.
- For more information: http://www.allaire.com

Option 7: PHP

- Personal Home Pages (PHP)
- An open source project written entirely by volunteers
- Provides simple, but powerful database access.
- Also great for rapid development.
- For additional information: http://www.php.net

Server Pages

- Microsoft IIS + ASP (<u>Active Server Pages</u>)
- Sun followed suit: JSP (<u>JavaServer Pages</u>)

JSP

- Uses XML-like tags and scriptlets written in the Java programming language to encapsulate the logic that generates the content for the page
- An extension of the Java Servlet API
- "Together, JSP and Servlets provide an attractive alternative to other types of dynamic web scripting/programming ..."

Servlet

- Java's answer to the Common Gateway Interface (CGI).
- <u>Applet</u>: a java program that runs within the web browser.
- <u>Servlet</u>: a java program that runs within the web server.
- Rapidly becoming the standard for building web applications.

Advantages of Servlets

- Servlets have six main advantages:
  - Efficient
  - Convenient
  - Powerful
  - Portable
  - Secure
  - Inexpensive

Advantage 1: Efficient

- For each browser request, the servlet spawns a light weight thread.
- This is faster and more efficient that spawning a new operating system process.
- Hence, servlets have better performance and better scalability than traditional CGI.

Advantage 2: Convenient

- Servlets include built-in functionality for:
  - Reading HTML form data
  - Handling cookies
  - Tracking user sessions
  - Setting HTTP headers
- Java is object oriented

4

Advantage 3: Powerful

- Servlets can talk directly to the web servers.
- Multiple servlets can share data:
  - Particularly important for maintaining database connections.
- Includes powerful techniques for tracking user sessions.

Advantage 4: Portable

- One of the advantages of Java is its portability across different operating systems.
- Servlets have the same advantages.
- You can therefore write your servlets on Windows, then deploy them on UNIX.
- You can also run any of your servlets on any Java-enabled web server, with no code changes.

Advantage 5: Secure

- Traditional CGI programs have a number of known security vulnerabilities.
- Hence, you usually need to include a separate Perl/CGI module to supply the necessary security protection.
- Java has a number of built-in security layers.
- Hence, servlets are considered more secure than traditional CGI programs.

Advantage 6: Inexpensive

- You can download free servlet kits for development use.
- You can therefore get started for free!
- Nonetheless, production strength servlet web servers can get quite expensive.

Java Server Pages

- Related to Java Servlets
- Can be used alone or in conjunction with servlets
- Represent (yet) another method for creating server side applications
- Unfortunately, we will not have time to cover the details of JSP.
- Nonetheless, I did want to give you a brief overview.

Servlets v. JSP

- Servlets
  - code looks like a regular Java program.
- JSP
  - embed Java commands directly within HTML
- Let's examine a Servlet program next to a JSP program...
- Each of these prints, "Hello, World!"

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {
  public void doGet(HttpServletRequest req, HttpServletResponse res)
  throws ServletException, IOException {
   res.setContentType("text/html");
   PrintWriter out = res.getWriter();
   out.println("<HTML>");
   out.println("<HEAD><TITLE>Hello World</TITLE></HEAD>");
   out.println("<BODY>");
   out.println("<BIG>Hello World</BIG>");
   out.println("</BODY></HTML>");
  }
}
```

A Java Servlet :
Looks like a regular
Java program

A JSP Page :
Looks like a regular
HTML page.

```
<html>
<head>
<title>Hello, World JSP Example</title>
</head>
<body>
 <h2> Hello, World!
The current time in milliseconds is
<%= System.currentTimeMillis() %>
</h2>
</body>
</html>
```

Embedded Java
command to
print current time.

6