Lecture 9: Internet Applications

CS178: Programming Parallel and Distributed Systems

February 26, 2001
Steven P. Reiss

I. Overview

A. Distributed Applications

1. **So far we have been dealing with client-server applications**
2. **The client-server model works and is widely used**
3. **But it is not without its problems**
   a) Especially when you look at today's applications

B. Client-Server problems

1. **Designed to work on a local network**
   a) Little provision for outside access
   b) Little hope of getting through firewalls
   c) Not much done for security/authentications
2. **Client is a real application**
   a) Has to be installed on each user's machine
   b) User has to have a machine capable of running it
   c) Problems with distributing new versions or working with old versions of the client
3. **Scalablility issues**
   a) You can design scalable client-server applications, but it is quite difficult and a lot of care needs to be taken
4. **Server and client are tightly integrated**
   a) Upgrading one implies upgrading the other

C. Suppose we want to go beyond

1. **To build applications for the internet**
2. **To have applications without these problems**
3. **Answer: Use the internet**

II. Basic Internet Application Architecture

A. Client

1. **Web browser on user's machine**
2. **Can be any browser on any machine**
3. **Machine can be anywhere**

B. Server

1. **Server still exists on base machine**
2. **Server still controls resources, databases, etc.**
3. **Server itself can be a complex, distributed application**

C. Middleware

1. **But server and client are not directly connected**
2. **Instead there is a large block of middleware -- code inbetween the server and the client**
3. **This is basically the web browser**
4. **But is extended with code that is applications specific**

D. Connections

1. **Middleware connects to client using the internet and HTTP**
   a) This is the current standard
   b) Allows communication through firewall, etc.
   c) HTTP is machine and source/target independent
2. **Middleware connects to server using the local net**

E. How does this solve the previous problems

1. **Designed to work on a local network**
2. **Client is a real application**
3. **Scalablility issues**
   a) Can scale in multiple ways
   b) Can scale through the middleware as well as backend
4. **Server and client are tightly integrated**

III. Flexability and Options in the Architecture

A. There are a variety of options here

1. **Each has pros and cons**

2. Each has proponents and opponents

B. Client -- web browser

1. Simple HTML needed for page

a) With FORMS for providing input back to server

```
<HTML>
<BODY>
<FORM METHOD="GET" ACTION="...">
<P>
Enter login name:
<INPUT TYPE="text" VALUE NAME="login" SIZE=25">
</FORM>
</BODY>
</HTML>
```

b) The Action indicates a web address

c) The result of completing the form is sent back to the web server at that address followed by '?' and a list of name=value paiers

d) In this case login=... (whatever user typed)

e) Forms allow for

(1) Text fields

(2) Selecting one of a set of items

(3) Buttons

(4) Checkboxes

(5) File selection

(6) Password entry

f) Hidden forms can be used to pass back static values

g) Pros and cons

2. Dynamic HTML using Javascript

a) Javascript is a programming language

(1) Untyped (dynamically typed using strings or ints)

(2) Not Java (or C or C++ ...)

(3) Some object facilities, but generally procedural

b) Can be tied to the web page

(1) Code triggered on various events such as pushing a button, mousing over an area, typing, clicking, ...

(2) Code can change values of forms, send requests back to the server

(3) In theory, code can change all the elements of a page

c) Example

```
<script>
var user_id = “default”
function setup() {
   user_id = getUser()
}
function getUser() {
   var c = document.cookie.split(“;”)
   for (var i = 0; i < c.length; ++i) {
      var a = c[i].split(“=”)
      if (a[0] == “User”) return unescape(a[1])
   }
   return ““
}
</script>
```

d) Advantages

(1) Powerful; can do a lot client-side

(2) Can communicate to server (both ways)

e) Problems

(1) Javascript isn’t quite the same on all browsers

(2) Capabilities, especially advanced change

(3) Changes made to the underlying page or forms tend not to work with some regularity

(4) You code is essentially public

3. Java Applets for client interface

a) An applet is a class that inherits from java.applet.Applet

(1) init(), start(), stop() methods called as needed by browser

(2) Code can contain arbitrary java classes, etc

(3) Code subject to Java security

b) Easy to incorporate into web page

```
<applet width=”500” height=”400”
   code=”edu.brown.cs.cs178.TestApplet”
   code_base=”http://www.cs.brown.edu/courses/cs178/”
   archive=”test.jar” >
Label if applet doesn’t appear
</applet>
```

c) Advantages

(1) Allows arbitrary code

(2) Can establish sockets to the host that the page is on (and thus talk to server directly)

d) Disadvantages

(1) Doesn't interact with browser that well -- sizing issues, refresh, etc.

(2) Java interaction with javascript and the html page is limited

(3) Provides a non-standard interface to the user

(4) Applets don't work on all browsers

(5) Applets might work differently on different browsers; e.g. different versions of Java

4. Plugins

a) It's also possible to write C/C++ code for the client as plug-ins

b) These can do anything

(1) Full access to machine; generally have good access to browser capabilities

(2) Implemented as shared library, etc.

(3) Standard calling sequences

c) Disadvantages

(1) Need a separate plug in for each browser-hardware combination -- not generally portable

(2) User's don't like because of security issues

C. Webserver (Middleware)

1. Issue -- how to handle requests sent by web page

a) Want to direct them to the back end server

b) Need to take input from back end and pass back to page

c) Web server itself doesn't do this, but extensions allow it to be done relatively easilty

2. CGI programming

a) There is one directory that the web server trusts commands in, //host/cgi-bin/
   (1) User can put arbirary commands there
   (2) Commands can be arbitrary binaries
   (3) PERL has primitives for interpreting args and passing back result in http format
   (4) C can be used (passed as command line arguments) with care

b) Advantages
   (1) Arbitrary language and code usable
   (2) Support for perl and other languages
   (3) Program can do load balancing, etc.

c) Disadvantages
   (1) Need to install into cgi-bin (security, etc)
   (2) Each time a command comes in, a new process is spawned
   (3) No easy way to maintain state

3. Java Servlets

a) This is similar to applets, except java here exists in the webserver
   (1) Code is loaded dynamically from trusted directory
   (2) Arbitrary java code is allowed

b) Java code implements javax.servlet.Servlet interface
   (1) Can inherit from GenericServlet
   (2) init(), destroy() methods for start/stop
   (3) service(ServletRequest,ServletResponse) for handling messages from web page

c) Need to have J2EE and a web server that supports servlets

d) Has the ability to maintain state or context for a particular user/web page
   (1) ServletContext object does this
   (2) Managed by the web browser, accessible to servlet

e) Advantages

(1) As long as you write java, you can handle arbitrary things

(2) Runs multithreaded, allowing multiple requests to be handled in parallel

(3) Maintains context, etc.

f) Disadvantages

(1) Code detached from web pages

(2) Need to have Java code generate lots of html (lots of messy text operations)

4. Java Server Pages

a) This is a way of combining servlets with html

(1) Can be used with servlets

b) ASP -- Microsoft's version, uses active X components rather than java (Visual Basic)

c) Essentially in the html you can insert arbitrary java code

(1) With access to all servlet features (context)

(2) Using %{ ... %}

(3) This code can generate strings that become part of the we page

d) Can be more complex in that you can essentially next html inside the java code

(1) Thus you can have java loops that include html to be generated

e) Advantages

(1) Good for simple uses of java to generate html

(2) Combines output with java code

f) Disadvantages

(1) You don't want to put complex code on the page

(2) You might need support classes, etc. anyway

5. Procedure call interface

a) Think of a web address as an object

(1) Then think of being able to send a request to that object

(2) Passing arguments, etc.

b) This is one of the things that .Net is all about

(1) Can easily create web pages with embedded C# code

(2) C# sets up the web interaction automatically

(3) C# provides a procedure call front end (the program doesn't know its invoking a web object)

c) Objects can have state, etc.