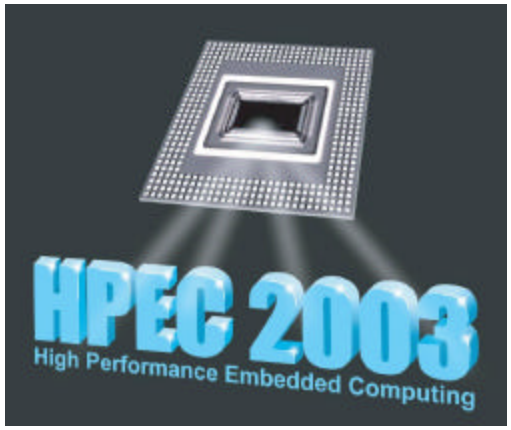


A Java based web interface to Matlab

Siddharth Samsi, Ashok Krishnamurthy, Stanley Ahalt, John Nehrbass, Marlon Pierce

Outline

- Motivation and Goals
- Matlab Web Server from MathWorks
- The OSU Matlab Application Portal
- Steps for creating a typical portal application
- An example Matlab application using the portal
- Advantages and Limitations
- Future work

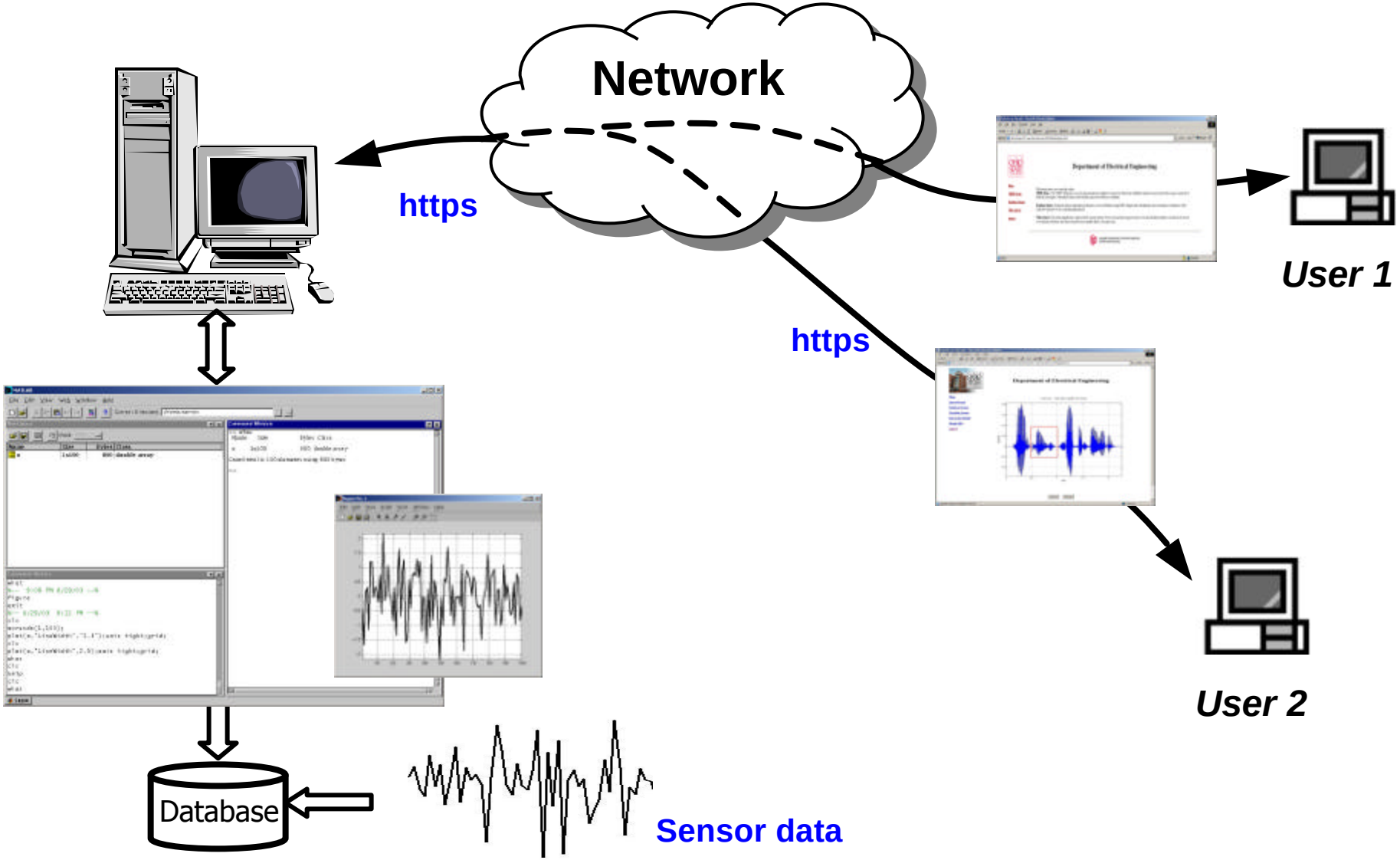
Motivations

- Matlab is a widely used computational environment for research and development
- Many large applications continue to be developed and deployed using Matlab
- Researchers from geographically distributed locations want to share applications and data
- Users want to run Matlab applications without having to buy licenses for all toolboxes
- Solution: A secure, web based Matlab application portal that allows researchers to upload code, run applications and visualize results

Goals

- Create a portal capable of running Matlab applications over the web
- Provide the ability to interactively zoom and examine 2-D and 3-D plots
- Provide the ability to upload Matlab code for testing and benchmarking on common data
- Provide secure access to the Matlab application portal through user authentication and encrypted communication

Goal

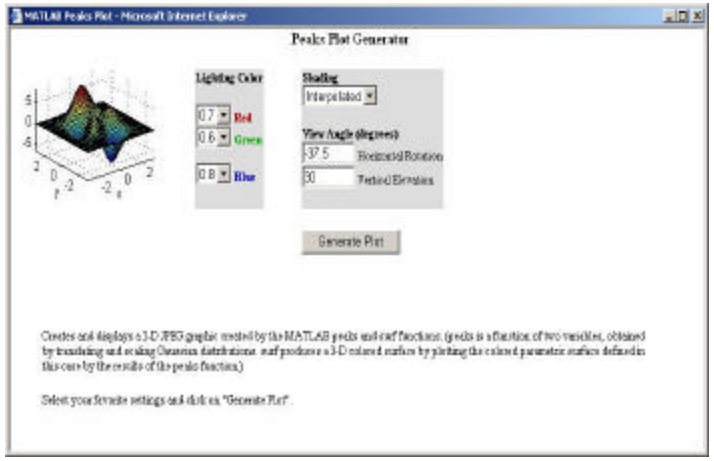


Benefits of Research Portal

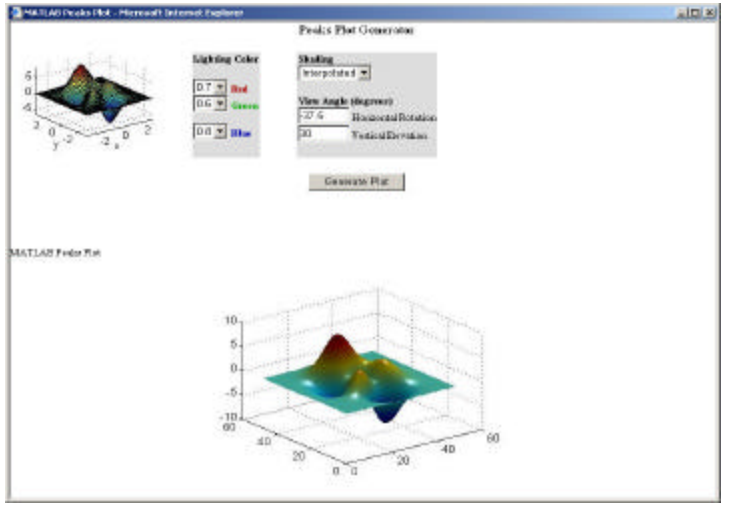
- Provides common platform for sharing data
- Enables easier sharing of code and results with the entire research community
- Single web based environment can provide easy access to all analysis tools
- Web interface can also be used to perform batch processing tasks more easily

Matlab Web Server

- Interface between the web and Matlab
- Uses Common Gateway Interface (CGI) to provide web based communication
- Provides helper functions for creating output HTML from result data
- Enables any Matlab application to be accessed over the web



Input Page



Results

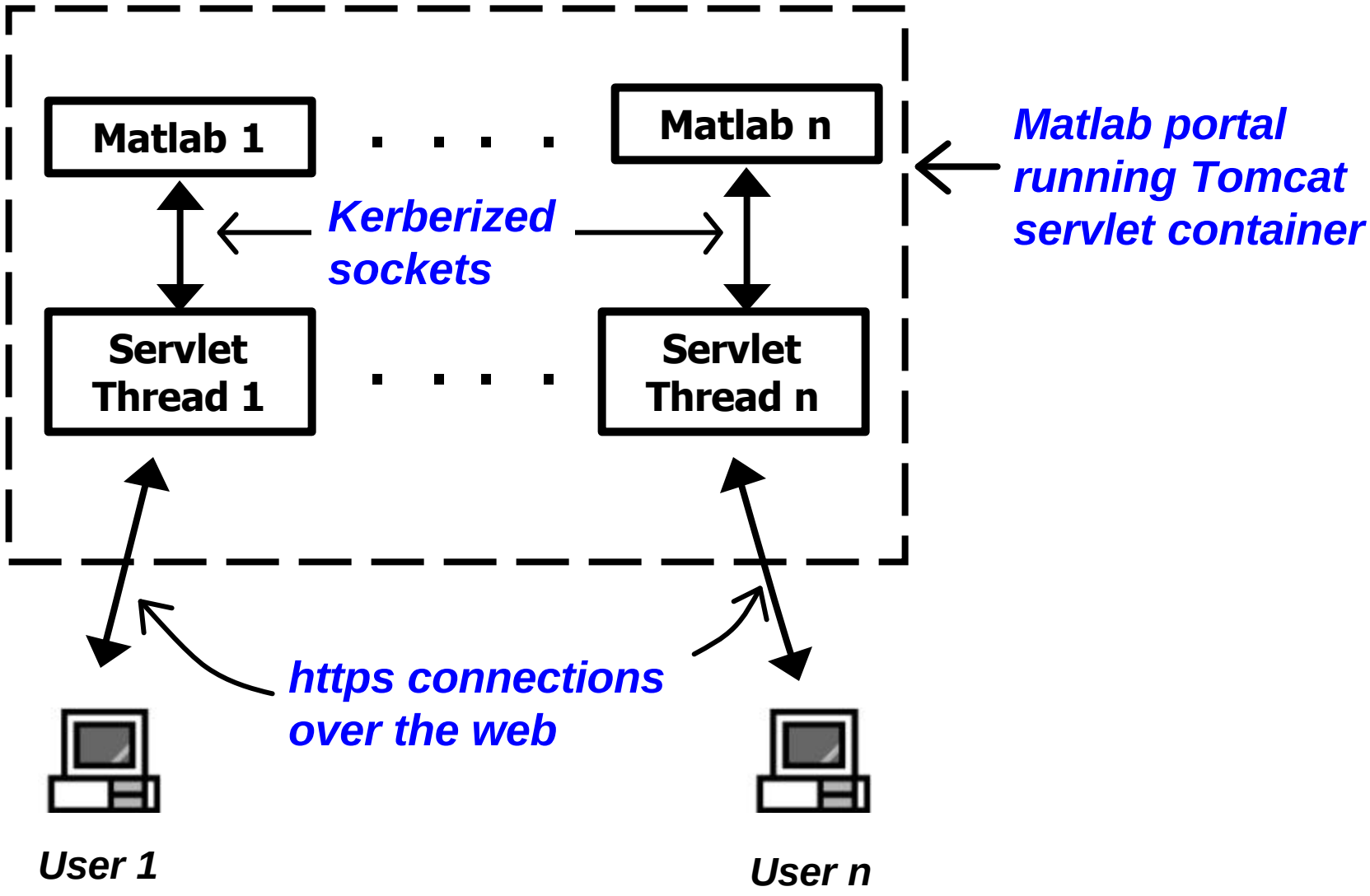
Limitations of Matlab Web Server

- Matlab workspace is not retained
  - All variables and data generated by an application is lost upon completion of the program
  - Results need to be recomputed for subsequent analysis by other applications
- Does not provide interactivity with Matlab graphical output
- Difficult to track users
  - The system does not have a concept of sessions
- Does not provide network security
  - No data encryption provided
  - User authentication not provided

The OSU Matlab Portal

- Based on standard open source industry components: Apache, Tomcat, Linux, MySQL
- Java technology used to build a web interface to Matlab
  - Matlab includes a Java Virtual Machine (JVM), providing access to Java objects
- Java servlets enable web based communication
- Kerberized Java sockets facilitate communication between Matlab an servlet
- Secure socket layer (SSL) used for encryption of all communication over the web

System Overview



Java Servlets

- Java servlets perform the following functions
  - Authenticate users
  - Start a new Matlab process for each new user
  - Communicate with user's browser using the https protocol
  - Communicate with user's Matlab process using kerberized sockets
  - Track users through sessions

Kerberos, Java Sockets and Servlet

- The Kerberos mechanism is used for secure message exchanges using sockets
- The OSU Matlab portal uses Kerberos V5 mechanism for secure communication
- Kerberos ticket required for establishing credentials and secure communication
- Kerberos tickets expire when user logs out of the portal

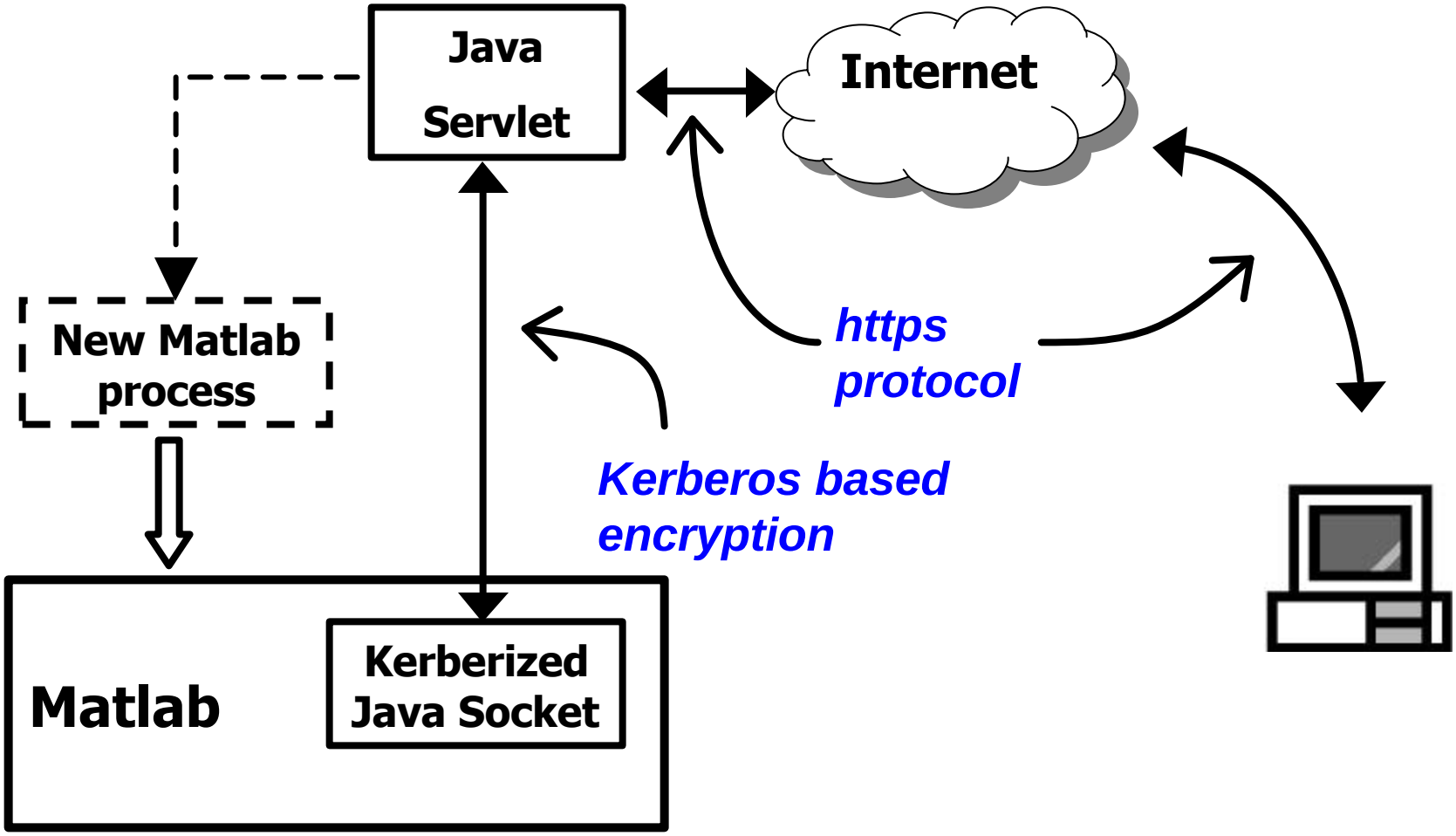
Kerberos based communication

- Once a user is authenticated and logged in, a unique Kerberos ticket is generated
- Communication process:
  - Connection is established between Java socket and Servlet
  - Socket and Servlet instantiate a new security context for communication
  - Using the Kerberos ticket the Socket and Servlet mutually authenticate and exchange tokens for encryption
  - All communication between them is now encrypted using previously exchanged tokens

Database Access using JDBC

- Matlab provides the ability to access Java objects
- Custom Java classes and Java Database Connectivity (JDBC) are used to accesses databases
- Matlab can extract data from JDBC compliant databases using these classes
- Provides independence from the Matlab Database Toolbox

User Login Process



Designing Portal Applications

- Create necessary Matlab m-files
- Input to Matlab
  - Obtained from user, over the web
  - HTML forms can be used
- Output from Matlab
  - Results are displayed in the user's browser
  - Necessary HTML can be created in Matlab as a string
  - Can use templates for generating output HTML
- Modification of configuration file
  - New applications should be registered with the server

Input from user to Matlab

- **Input parameters to Matlab obtained from the web**
- **Standard HTML forms can be used to obtain user input**
- **HTML forms support input in the form of**
  - **Plain text**
  - **Pull down menus**
  - **Boolean operators (e.g.: HTML Radio buttons)**

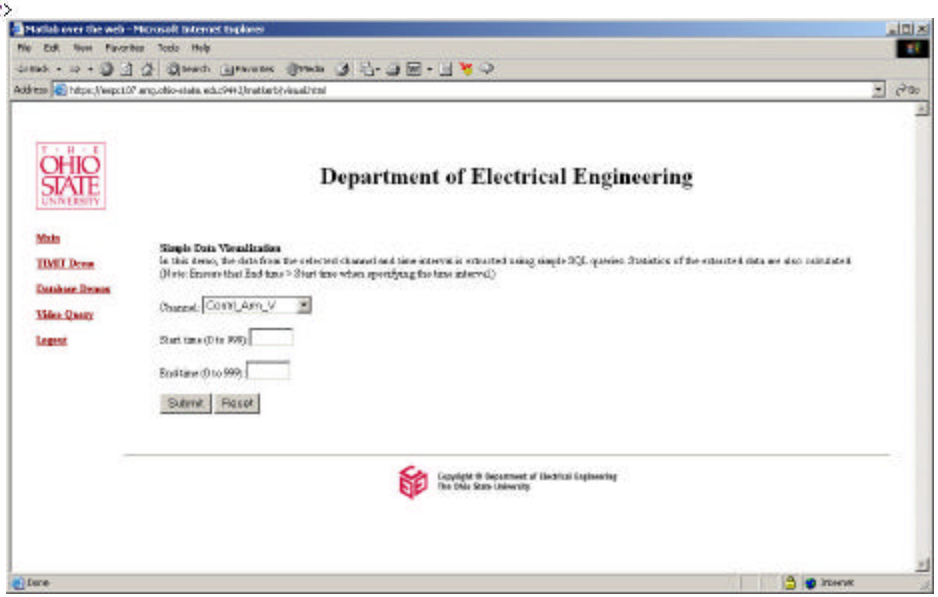
Sample Input HTML

```
<form method="POST" action="/matkerb/clientServlet">
Channel: <select name="channel">
<option value="1702">   Contrl_Arm_V
<option value="1703">   Lat_Accel
<option value="1704">   Brake_ON
<option value="1705">   Inter_Accel_V
<option value="1706">   Rear_Accel_V

</select>
<br><br>

Start time (0 to 999):<input type="text" size="5" maxlength="4" name="start"><br><br>
End time (0 to 999):<input type="text" size="5" maxlength="4" name="stop"><br><br>

<input type="hidden" name="name" value="dbvisual">
<input type="hidden" name="exit" value="-1">
<input type="submit" value="Submit" name="name channel start stop">
<input type="reset" name="Reset">
```

Call servlet to send request to Matlab

Name of application m-file to run

List of parameters for the specified application m-file



Output from Matlab

- **Results from Matlab are displayed in the user's browser**
- **Output HTML can be created as a Matlab string**
  - **Application m-file responsible for adding necessary HTML tags**
  - **Data from Matlab should be converted to appropriate form**
- **Use of HTML templates**
  - **Helper functions are provided to substitute the appropriate data into a template**
  - **Simpler to change the way results are displayed**

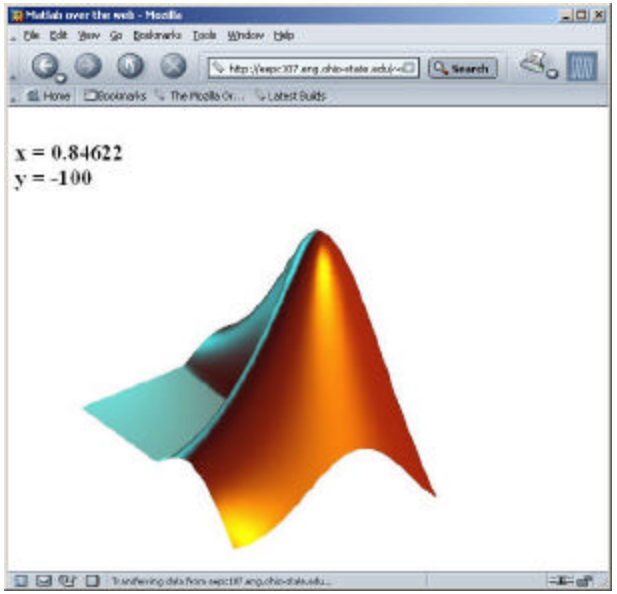
Sample Output HTML using templates

```
<html><head><title>Matlab over the web</title>
</head><body>
<br>x = #x#
<br>y = #y# <br>
<image src="#image1#">
</body></html>
```

The '#' sign is used to enclose the output variables in HTML template

- Use Matlab function *gethtml,* for creating the output HTML :
  html = gethtml ('template.html', 'x', randn(1), 'y', -100, 'image1', 'logo.jpg');

```
<html><head><title>Matlab over the web</title>
</head><body>
<br>x = 0.84622
<br>y = -100 <br>
<image src="http://eepc107.eng.ohio-state.edu/logo.jpg">
</body></html>
```



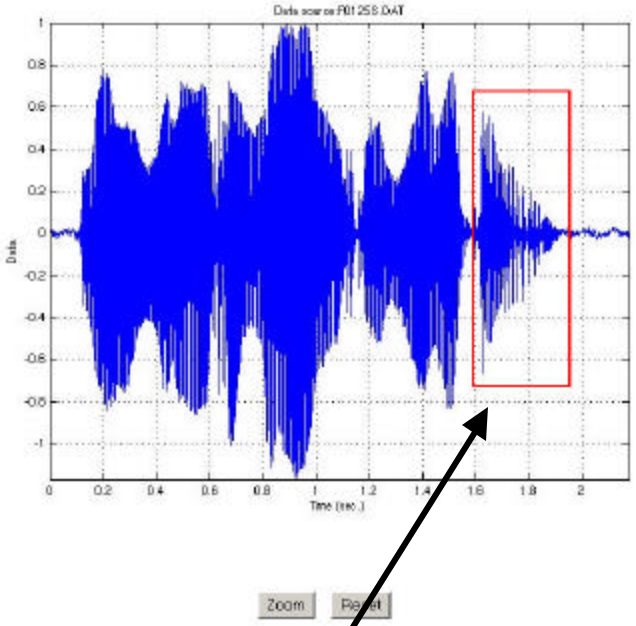
Interacting with Matlab graphics

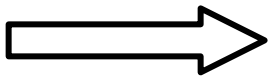
- Java Applets are provided to enable interaction with Matlab generated graphics
- Applets facilitate:
  - Displaying of images at desired location
  - Capturing mouse events and mouse pointer co-ordinates
  - Drawing lines and rectangles to show the “zoom area”
- Applets also give the ability to play .wav and .au files

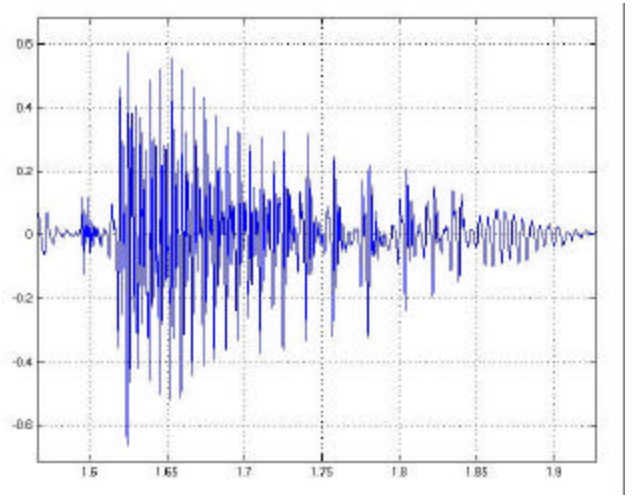
Interacting with Matlab graphics

- JavaScript is used to access mouse co-ordinates from the Java Applet
- Used to set parameter values to be sent to Matlab
- JavaScript can also be used to generate web pages

Example of graphical interaction







- “Zoom” area drawn by Applet.
- Zoom co-ordinates are read using JavaScript and sent to Matlab

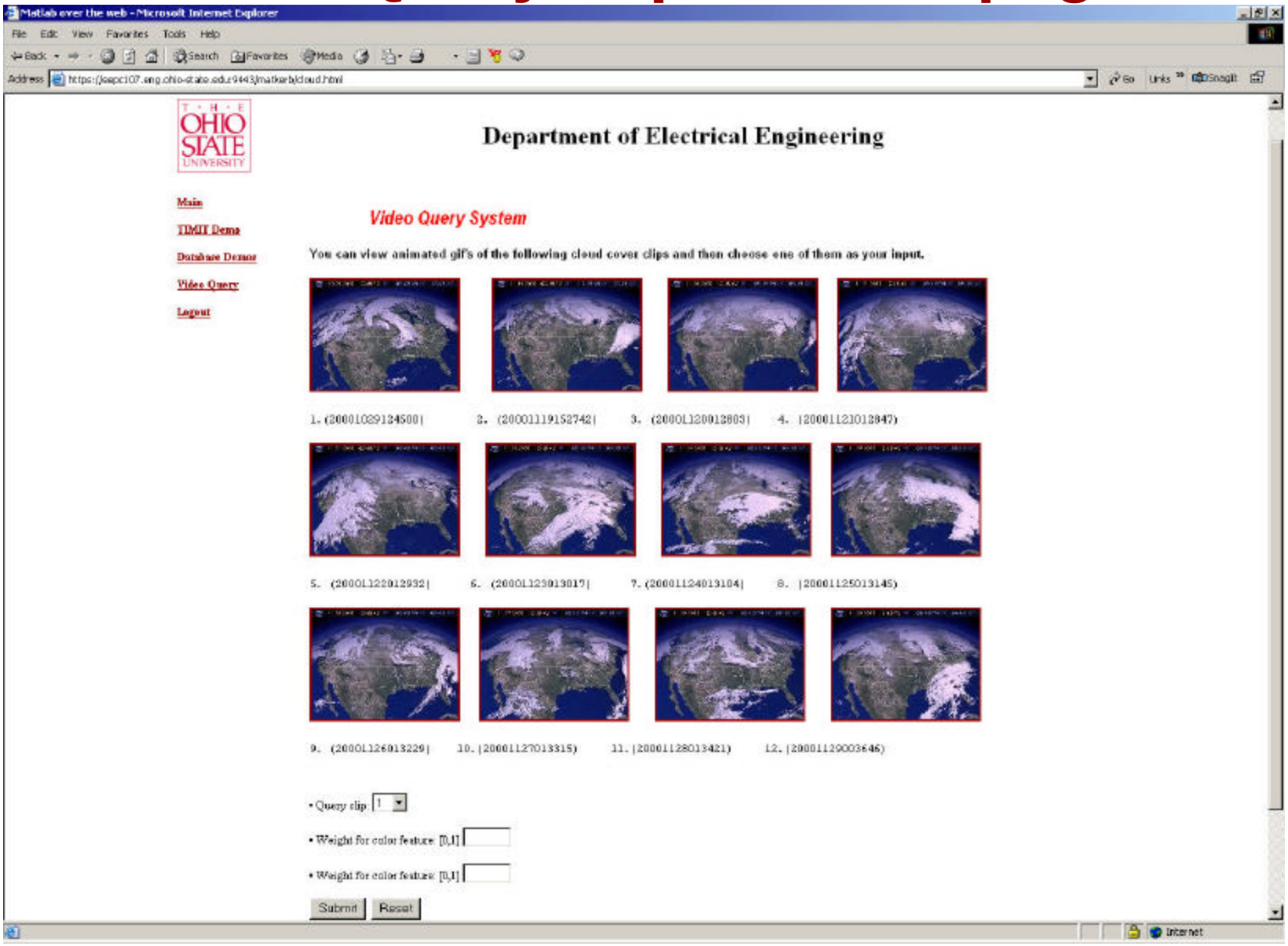
Result of zoom

- Zooming achieved by replotting data with appropriate axes

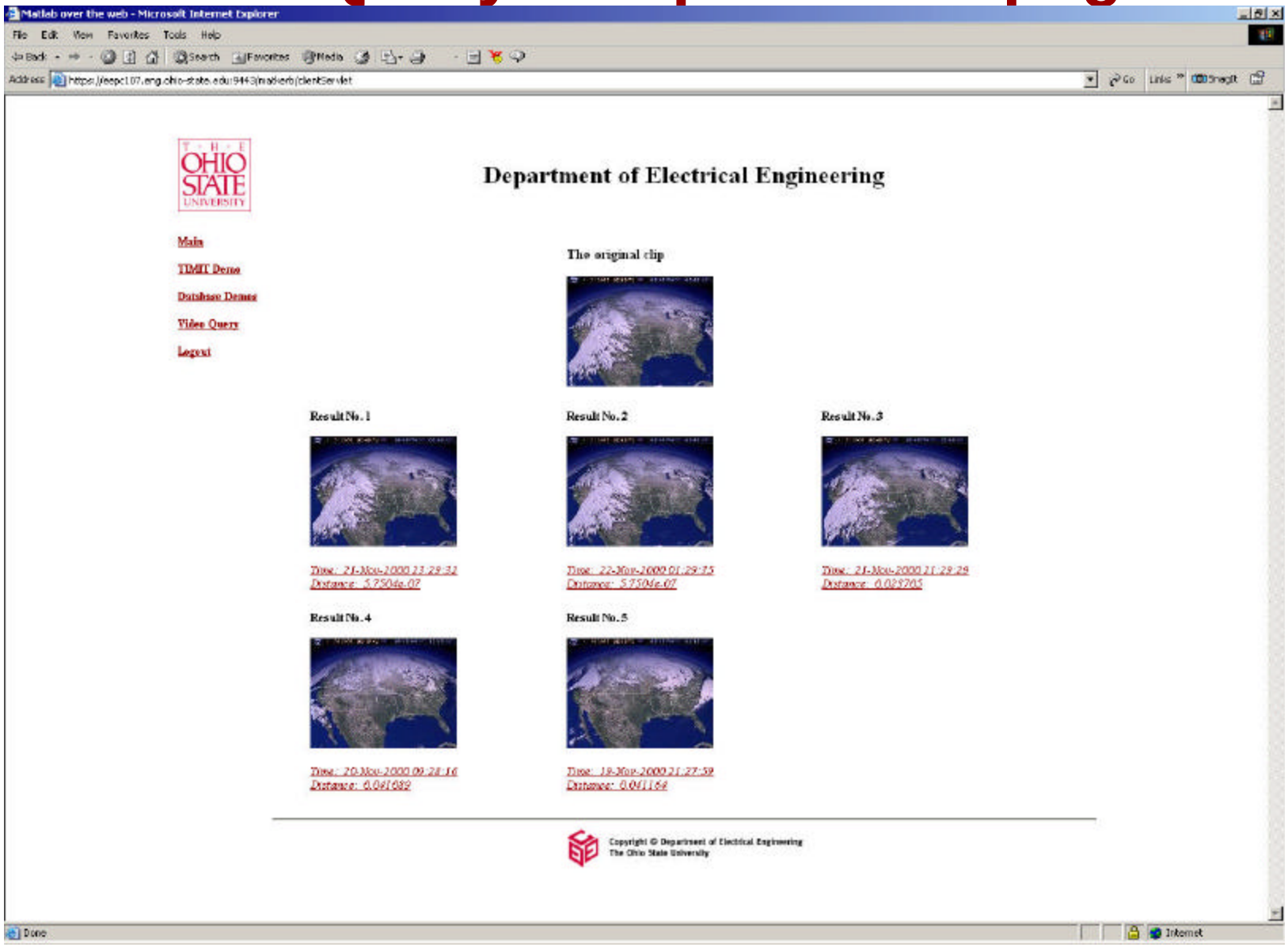
Sample Application

- **Video Query System:**
  - **User selects an input cloud cover image**
  - **Available cloud cover images are in the form of animated gifs**
  - **User chooses the weights to be assigned to the color feature and the motion feature**
  - **Based on user input, Matlab returns the images most similar to the test image**

Video Query: Input HTML page



Video Query: Output HTML page



Deploying OSU Matlab Portal

- **Core OSU Matlab components**
  - **Java Servlets and classes**
  - **Matlab m-files**
  - **Basic HTML web pages**
- **Software and Libraries required**
  - **Virtual Network Computing (VNC): Needed for providing Matlab with a virtual X-display**
  - **Kerberos clients**
  - **Tomcat servlet container**
  - **Java SDK**
  - **Apache Ant: Needed for compilation of Java source**
  - **C compiler**

Installing the OSU Matlab Portal

- **Shell scripts are provided to aid the installation of the portal**
- **Shell script performs following functions:**
  - **Creating the necessary directory structure**
  - **Reading environment variables and modifying the Java source code accordingly**
  - **Compilation of all source code**
  - **All class files, Matlab m-files and configuration files are put into appropriate directories**
  - **Cleanup: Removal of intermediate files**

Registering Portal Applications

- **Any new Matlab application to deployed must be registered with the portal**
- **For this, a configuration file is provided: *deploy.xml***
- ***deploy.xml* contains:**
  - **Application name**
  - **Input parameters expected and their names**
  - **Database to used, if needed**
  - **Application m – file name**

Sample Configuration file *deploy.xml*

```
<?xml version="1.0"
   encoding="ISO-8859-1" ?>
<webapps>
<application>
<name> dbvisual </name> }
<num - args>2</num - args>
<args>
<param>channel</param>
<param>stop</param>
</args>
<db>
<name>atcdata </name> }
</db>
<mfile> dbvisual</mfile> }
</application>
</webapps>
```

Application name

List of input parameters

Database to be used

Application m-file

Limitations

- **Matlab memory requirements**
  - **Each Matlab process uses 50 Mb RAM on startup**
  - **Memory used increases as more variables are created**
- **JavaScript and Applets needed for interacting with graphics**
  - **Disabling JavaScript removes all graphics interactivity**
- **Currently limited to Unix/Linux platforms**

Future Work

- Provide ability to upload and download data as well as Matlab code
- Provide bulletin board for exchange of ideas, problem discussion, etc.
- Develop administrator tools for portal
  - Make provisions for adding/removing users
  - Tools for portal administration
- Provide comprehensive documentation for the portal

Conclusion

- The OSU Matlab Portal has more flexibility than the Matlab Web Server from MathWorks
- Possible to create more interactive applications, e.g: zooming into images
- Eliminates the need for each user to buy Matlab and all toolboxes
- Less expensive alternative since it is based on freely available software/libraries

Additional Slides

Matlab Display Issues

- When run as a background process, Matlab runs in the terminal emulation mode
- Problems:
  - No X-Display available for Matlab
  - Cannot produce JPEG images directly
  - Representation of result data severely limited
- Solution:
  - Use Virtual Network Computing (VNC)
  - Matlab uses this X display for generating graphics

Virtual Network Computing (VNC)

- **Remote display system**
- **Used to create a virtual desktops**
- **This virtual desktop can be access from a variety of platforms (Unix/Linux, Windows, MacOS)**
- **Has very small memory requirements**
- **Web site:**

  http://www.uk.research.att.com/vnc/

Input to Matlab from the Web

- **All Matlab applications deployed through the portal get input parameters from the user over the web**
- **Names of parameters are specified in *deploy.xml***
- **The parameters are returned in the form of a structure *paramStruct***
- **All applications have access to this variable in the workspace**
- **Applications must convert parameters from a string to appropriate format**

Example: Accessing input parameters

- **Consider an application with the following input parameters:**
  - ***channel*** **: string**
  - ***start_time*****: integer**
  - ***stop_time*****: integer**
- **User input over the web:**
  - ***channel='engine_speed'***
  - ***start_time=10***
  - ***stop_time=40***

Example (continued)

- **In Matlab, the parameters can be accessed as:**

channel_name = paramStruct.channel ;

start = str2num (paramStruct.start_time) ;

end = str2num (paramStruct.stop_time) ;

- **All parameters obtained over the web are available are accessed as strings in Matlab**

Java Socket used by Matlab

- **Matlab uses Java sockets for communicating with the Servlet**
- **Data obtained over the web and results to be sent back to the browser are obtained by Matlab using this socket**
- **This socket uses Kerberos for secure communication with the Servlet**

Sending Results to Browser

- **Applications are responsible for generating HTML required to display results in the user's browser**
- **Applications need only create the HTML in the form of a variable named *html***
- **This string will be sent back to the user's browser by the Java socket**

Generating Images from Matlab plots

- **Images to be displayed in the browser should be in the JPEG format**
- **The Matlab *'print'* function is used to print figures to JPEG images**
- **Helper functions are provided for creating filenames, generating JPEG images and creating necessary HTML tags for displaying images**

Example: Creating and Displaying Images

- **Following code illustrates the use of helper function for creating images:**

```
y = linspace(-2*pi,2*pi) ;
x = exp(y).*cos(y) ;
figure ;
handle = plot(x,y) ; axis tight ; grid ;
file_name = getFileName ;
makejpeg (file_name, handle) ;
html = addImage ( file_name ) ;
```

getFileName **: Returns a randomly generated filename**

makejpeg **: Prints the figure to JPEG format**

addImage **: Creates HTML <img> tags with appropriate file name and path to file**