

A Java-based Web Interface to Matlab
Siddharth Samsi, Ashok Krishnamurthy and Stanley Ahalt
The Ohio State University

Matlab, from The Mathworks Inc., is a technical computing environment that is widely used in the scientific and engineering communities for research and development. The basic product can be augmented by a variety of toolboxes that extend its capabilities in specialized areas such as Signal and Image processing, Control System Design, Financial Analysis, etc. There are many situations in which it is desirable to access MATLAB and toolbox functionality over the World Wide Web. For example, a research group may want to allow other researchers to test and benchmark their MATLAB code using data available over the web. The Matlab Web Server, also available from MathWorks Inc., makes it possible to deploy Matlab applications over the internet. However, the Matlab Web Server does not preserve the user workspace, making it difficult to create applications that make the online Matlab experience as rich as a locally run Matlab session.

This paper discusses the development of an alternative method for deploying Matlab applications over the Web. The system developed here aims at overcoming many of the limitations of the Matlab Web Server, resulting in a more interactive online Matlab experience. Using the MATLAB-Java interface available in the recent releases of MATLAB, our system provides the web interface through the use of Java Servlets and custom Java classes. As shown in Fig. 1, a multithreaded socket is used to start a new Matlab process for every user that logs into the system. Once the Matlab process for a user has been started, all communication between the user and Matlab process is facilitated by the servlet and the Java socket opened by Matlab. This allows each user to have a workspace that is preserved until his/her Matlab process exits. This also has the advantage that no additional server-side programming is needed specifically to save user data and have a unique identifier to associate it with the user.

We will show an example of a web-accessed MATLAB application that allows a user to access acoustic signals stored in a mySQL database, search for signals that meet a user-selected criterion, analyse the selected signal using a variety of signal processing algorithms and create 2-d and 3-d plots of the signal and analysis results. The user also has the ability to playback and listen to selected portions of the signal. The preservation of the user workspace means that the user can zoom in and examine portions of the signals in more detail without having to rerun the analysis each time. The zooming and playback of the signals is accomplished using Java applets that are downloaded to the user's browser. This example

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE 20 AUG 2004		2. REPORT TYPE N/A		3. DATES COVERED -	
4. TITLE AND SUBTITLE A Java based web interface to Matlab				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) The Ohio State University Columbus, OH 43210				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, distribution unlimited					
13. SUPPLEMENTARY NOTES See also ADM001694, HPEC-6-Vol 1 ESC-TR-2003-081; High Performance Embedded Computing (HPEC) Workshop (7th)., The original document contains color images.					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

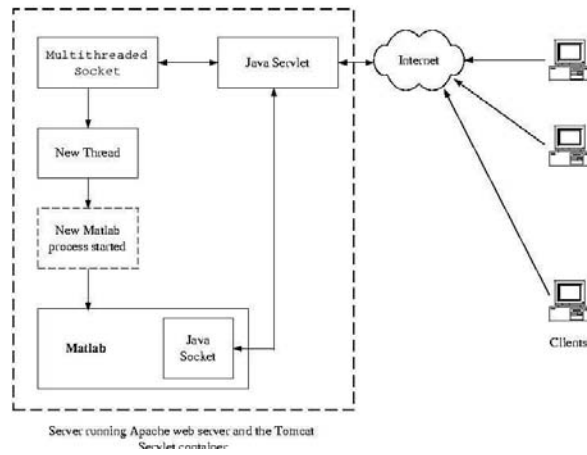


Figure 1: System overview

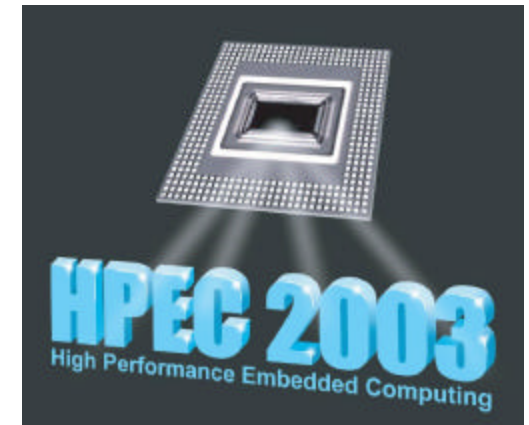
system also shows how to use Java Database Connectivity (JDBC) to access and manipulate the signal database.

An important application of such a system is a research oriented web portal where a community of scientists and researchers can share data and code with each other. The portal can allow authenticated users to download MATLAB code which is then immediately available to other users to test and examine results. Any specialized toolboxes required need be hosted only on the server system running MATLAB and need not be replicated at each user site.

This work was supported in part by the DoD High Performance Computing Modernization Program's PET program.

References

1. J. Hunter and W. Crawford, *Java Servlet Programming*, O'Reilly & Associates, 2 ed., 2001
2. D. R. Callaway and D. Coward, *Inside Servlets: Server-Side Programming for the Java(TM) Platform*, Addison-Wesley Pub. Co., 2nd ed., 2001
3. W. R. Stevens, *UNIX Network Programming*, Prentice Hall PTR, 2 ed., 1998
4. MATLAB and MATLAB Web Server documentation from MathWorks, Inc., <http://www.mathworks.com>
5. Sun Microsystems, <http://java.sun.com>
6. The Apache Software Foundation, <http://www.apache.org>
7. The Jakarta Project, <http://jakarta.apache.org>



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

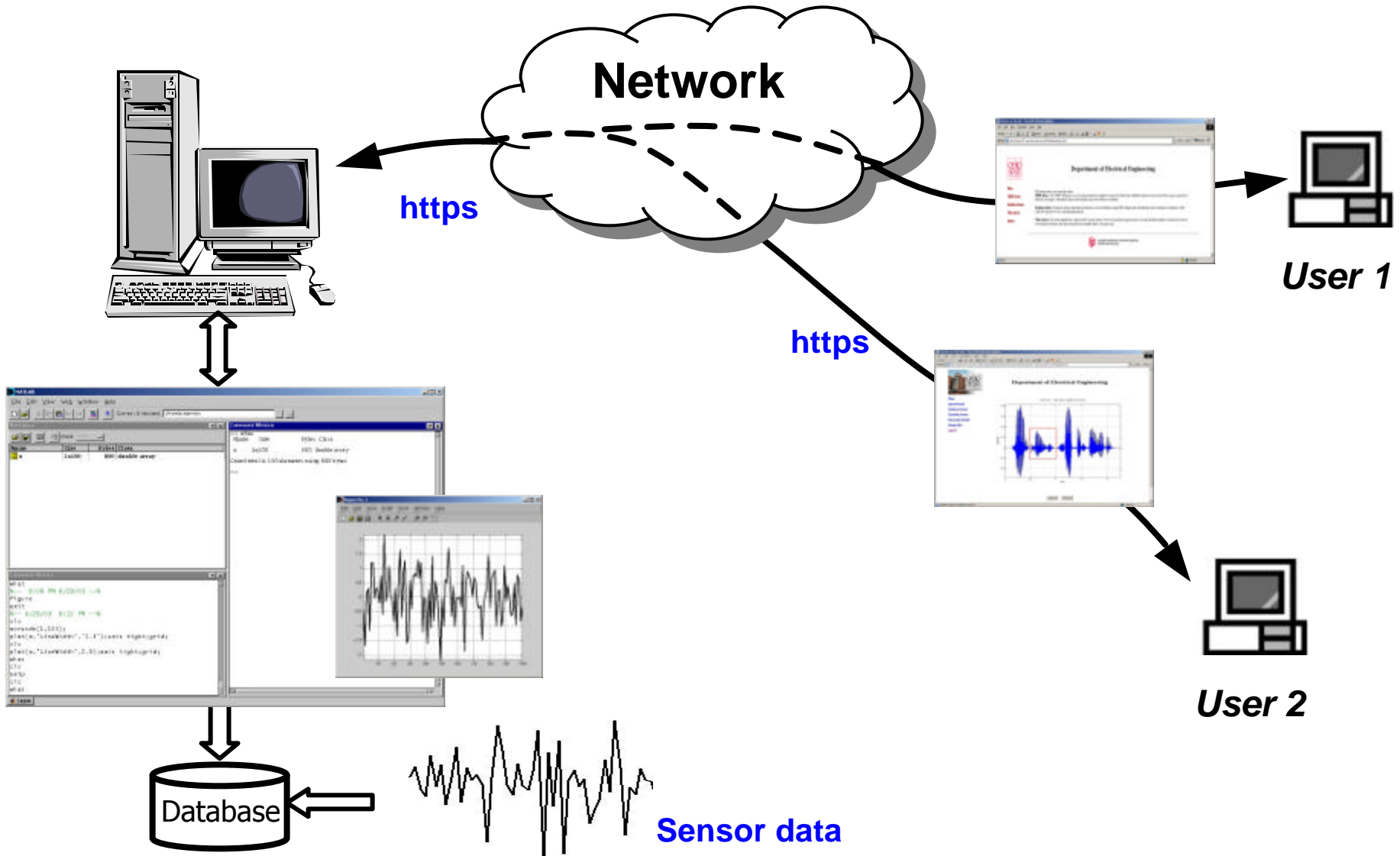
- **Matlab is a widely used computational environment for research and development**
- **Many large applications continue to be developed and deployed using Matlab**
- **Researchers from geographically distributed locations want to share applications and data**
- **Users want to run Matlab applications 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



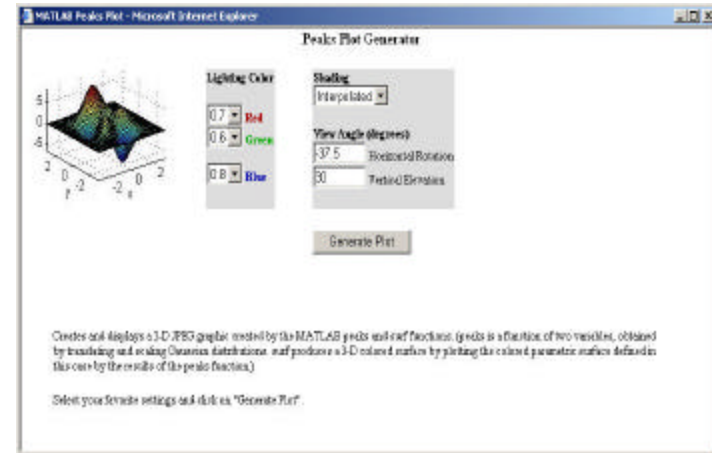
Benefits of Research Portal

- **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**

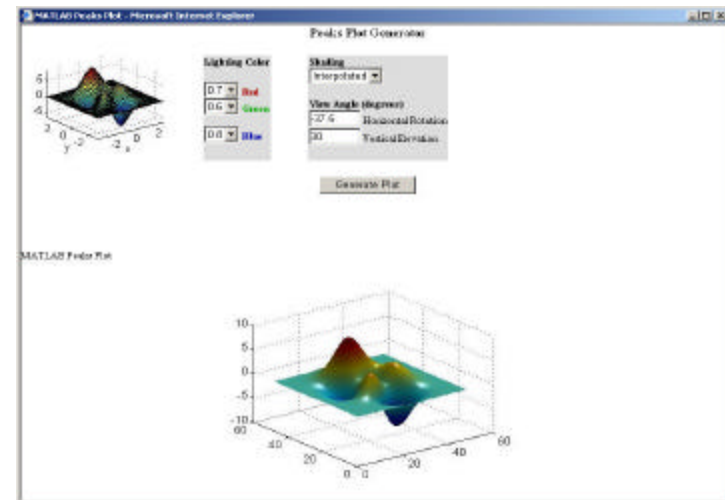


Matlab Web Server

- 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

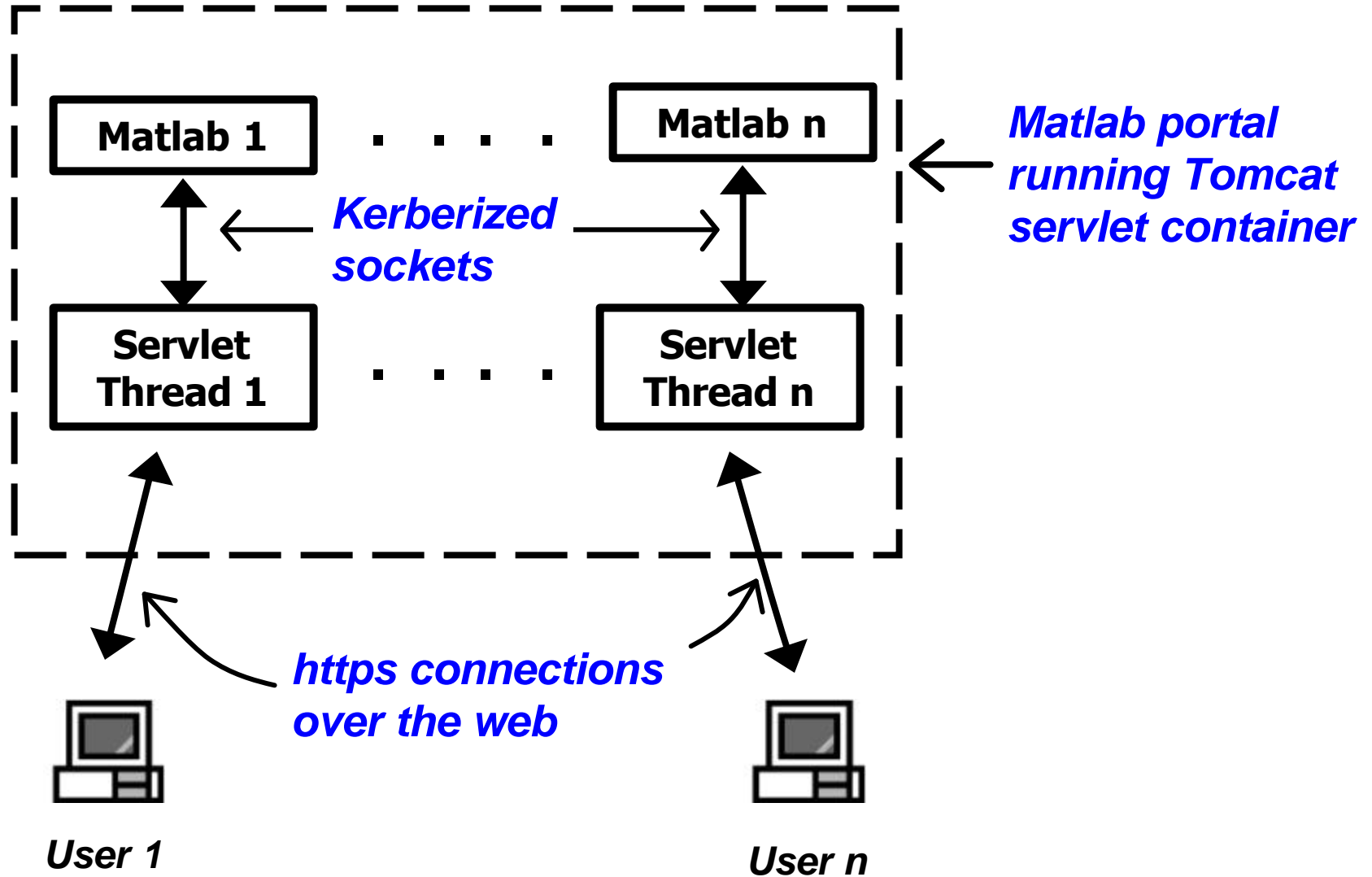
- **Matlab workspace is not retained**
 - All variables and data generated by an application is lost upon completion of the program
 - Results need to be recomputed for subsequent analysis by other applications
- **Does not provide interactivity with Matlab graphical output**
- **Difficult to track users**
 - 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**

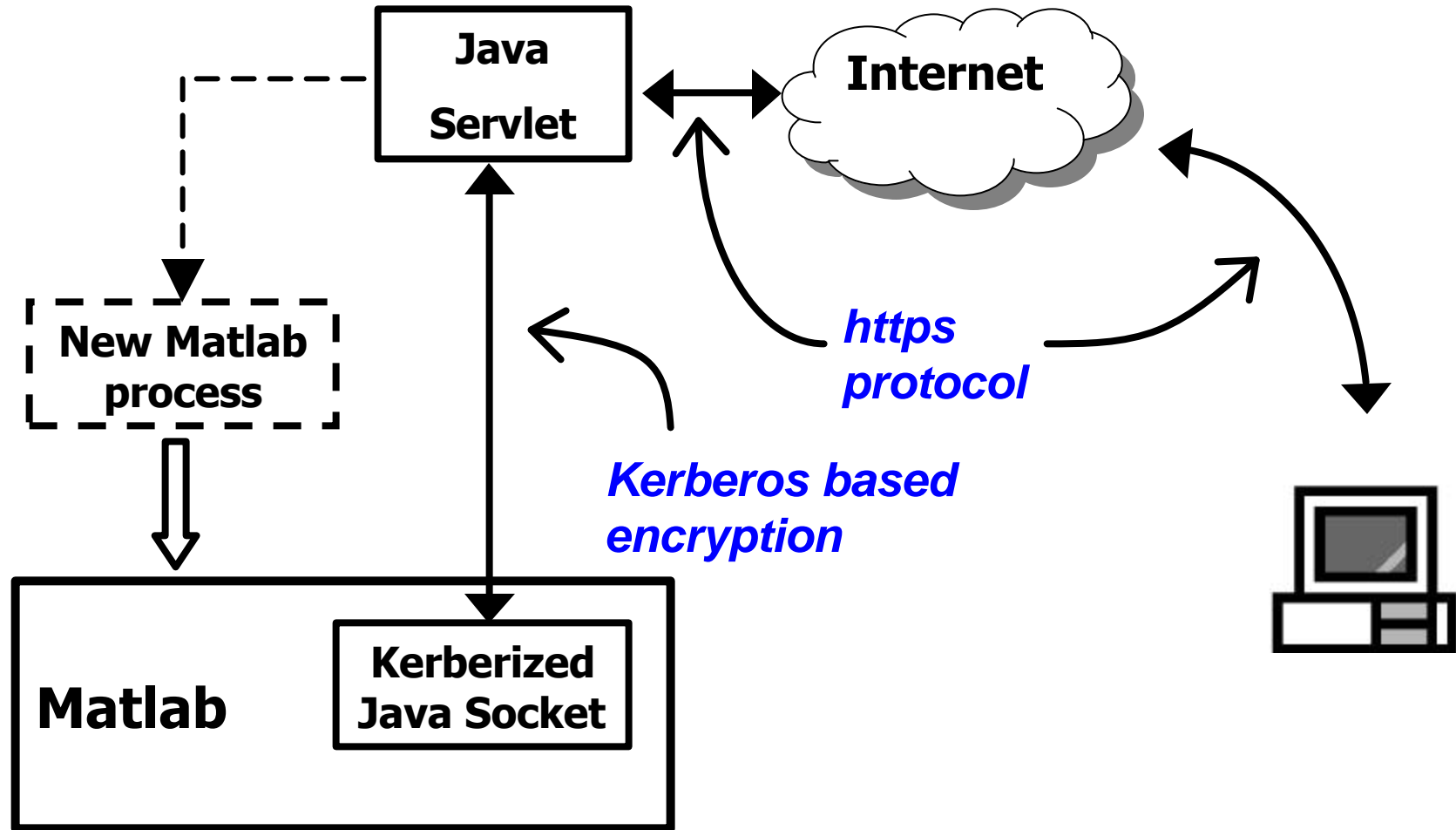


Database Access using JDBC

- **Matlab provides the ability to access Java objects**
- **Custom Java classes and Java Database Connectivity (JDBC) are used to access 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)



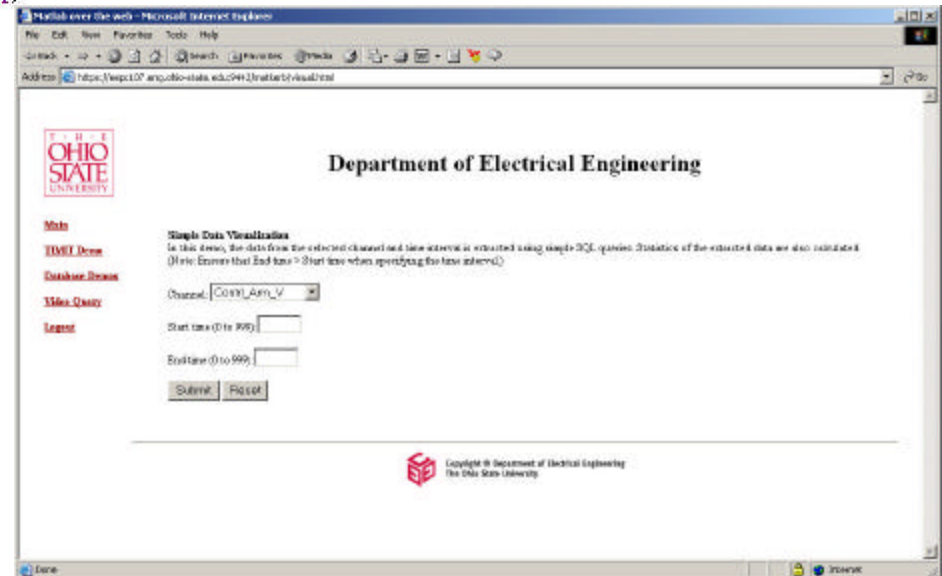
Sample Input HTML

Call servlet to send request to Matlab

```
<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">
```

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



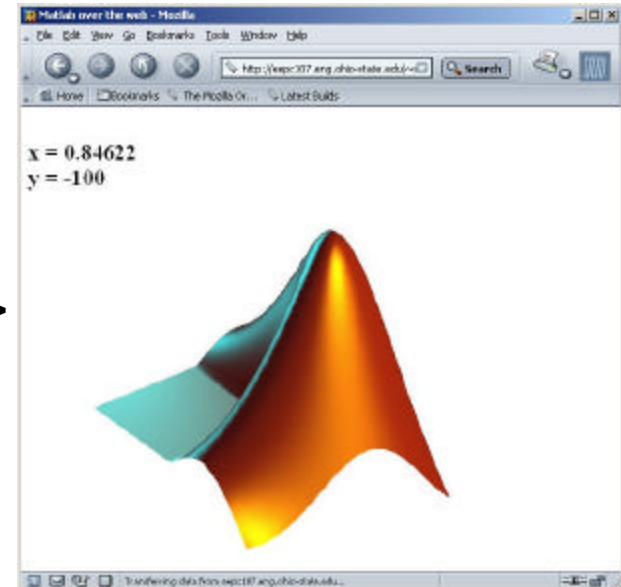
Sample Output HTML using templates

```
<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 coordinates**
 - **Drawing lines and rectangles to show the “zoom area”**
- **Applets also give the ability to play .wav and .au files**

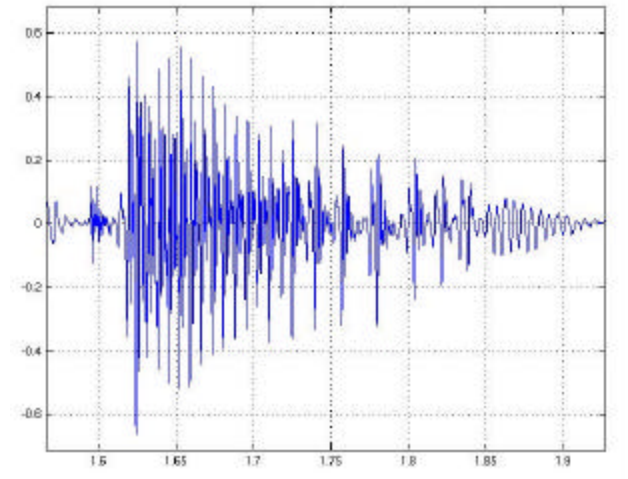
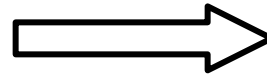
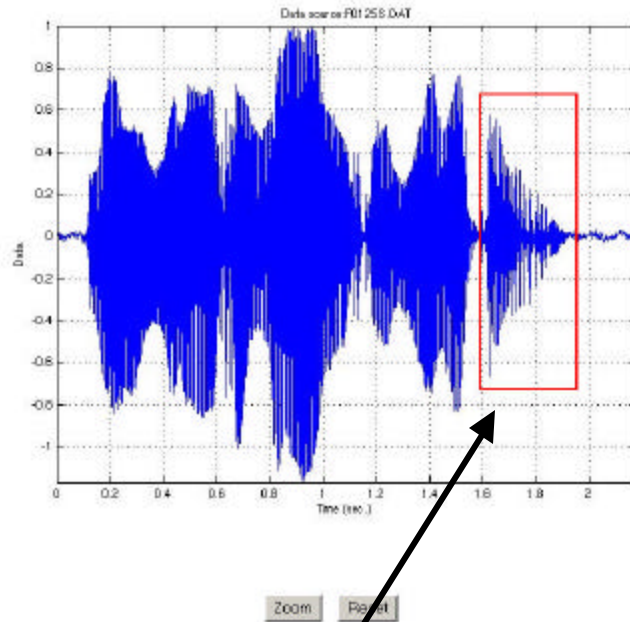


Interacting with Matlab graphics

- **JavaScript is used to access mouse coordinates 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



Result of zoom

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

- 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






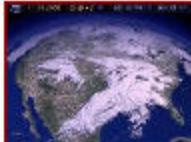
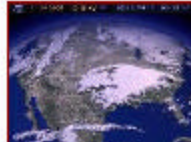
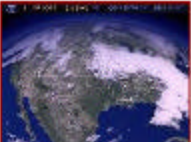



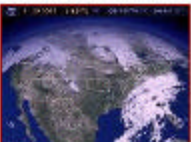
Microsoft Internet Explorer
Address: https://joepc107.eng.ohio-state.edu/~naik/b/cloud.html

 **Department of Electrical Engineering**

[Main](#)
[TDMT Demo](#)
[Database Demo](#)
[Video Query](#)
[Logout](#)

Video Query System

You can view animated gifs of the following cloud cover clips and then choose one of them as your input.

			
1. (20001029134500)	2. (20001119152742)	3. (20001120012803)	4. (20001121012847)
			
5. (20001122012932)	6. (20001123013017)	7. (20001124013104)	8. (20001125013145)
			
9. (20001126013229)	10. (20001127013315)	11. (20001128013421)	12. (20001129003646)

• Query clip:

• Weight for color feature:

• Weight for color feature:

Video Query: Output HTML page

The screenshot shows a Microsoft Internet Explorer browser window displaying a web page. The address bar shows the URL: <https://eeepc107.eng.ohio-state.edu/9443/malorb/clientServlet>. The page content includes:

- Navigation Menu (Left):**
 - [Main](#)
 - [TDMT Demo](#)
 - [Database Demo](#)
 - [Video Query](#)
 - [Logout](#)
- Department of Electrical Engineering** (Centered Title)
- The original clip** (Section Header) with a satellite image of North America.
- Search Results:**
 - Result No. 1:** Time: 21-Mar-2000 23:28:32, Distance: 5.7504e-07
 - Result No. 2:** Time: 22-Mar-2000 01:29:25, Distance: 5.7504e-07
 - Result No. 3:** Time: 21-Mar-2000 21:28:28, Distance: 0.029705
 - Result No. 4:** Time: 20-Mar-2000 09:28:16, Distance: 0.061692
 - Result No. 5:** Time: 19-Mar-2000 21:27:59, Distance: 0.061158
- Footer:** Copyright © Department of Electrical Engineering, The Ohio State University.

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 be 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 be 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> }
```

Application name

```
<num - args>2</num - args>
```

```
<args>
```

```
<param>channel</param> }
```

List of input parameters

```
<param>stop</param>
```

```
</args>
```

```
<db>
```

```
<name>atcdata </name> }
```

Database to be used

```
</db>
```

```
<mfile> dbvisual</mfile> }
```

Application m-file

```
</application>
```

```
</webapps>
```

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 tags with appropriate file name and path to file