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 it's 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-accesed 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

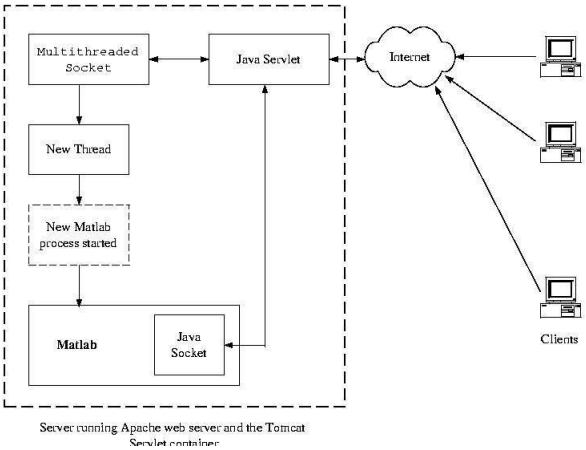

Figure 1: System overview

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 MathWotks, 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*