On-line Mathematical Utilities for Java Enabled Devices

Damir Krstinic, B.Sc. and Ivan Slapnicar, Ph.D.
Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture
University of Split
R. Boskovica b.b., 21000 Split, Croatia
E-mail: damir.krstinic@fesb.hr, ivan.slapnicar@fesb.hr
Phone: +385 21 305 617

Abstract— **We present mathematical utilities for solving and visualization of engineering problems. Since these tasks can be computationally very expensive, user is limited with available resources. This is for example the case for engineers working in the field conditions. By performing all of the calculation on the server side and providing user interface for different platforms, wide range of complicated scientific and engineering problems can be solved using cheep terminal equipment.**

We have developed application/service which brings Matlab compatible programming language functionality to java enabled mobile phones and similar devices. User can enter sequence of commands (short program) which are processed on the server side where all of the computation is done. Programming language used has high compatibility to Matlab and has functionality of graphical representation of the numerical results (drawing images). Application has a special module for drawing different types of functions, including 3D functions.

I. Introduction

WHILE scientists and engineers working in the labs usually have computational power they need for theirs experiments and calculations, engineers working in the filed conditions are limited by computational power of the portable devices available. Todays high-end portable computers are comparable with PC desktop workstations, and can solve a wide range of computational demanding tasks, but some problems require stronger workstations, including multiprocessor systems. Another limiting factor can be installed software. Beside that, portable computer is not always available, especially when engineer must solve unexpected problem without preparation and required equipment. For some users, like students, high-end portable computers may not be affordable.

By performing memory and computationally expensive calculations on strong servers, and providing interfaces for different mobile platforms, computational power can be brought to users without need for expensive and sometimes unpractical portable computers.

We have developed application/service *MathGSM* with two modules. Module *GSMLab* brings Matlab compatible programming language functionality to java enabled mobile devices like GSM mobile phones, handheld devices, etc. User is provided with interpreter for high-level programming language, primarily designed for numerical computations, which can be used to solve wide range of engineering problems. This way, user is not limited to set of problems specific software can solve. Mobile device is used only as the I/O terminal, and all of the computation is performed on server side.

Based on this service, number of specific problems solver applications can be developed for users who do not need (and probably do not know how to use) fully functional programming language, but want to have easy-to-use interface with well defined input values for their specific problem. Module *GSMPlot* is example of this approach. *GSMPlot* is designed for drawing different types of functions, including 3D functions.

Modules *GSMLab* and *GSMPlot* are wireless versions of the on-line web-based services Octave On-line *http://bolek.csc.unist.hr/czr/en/on-line.php3* and NetPlot *http://lavica.fesb.hr/netplot/en/* [1].

II. Technical requirements

IN creating this application we have used several tools, all of which are freely available on the Internet. On our servers we are using SuSE Linux 7.1 (*http://www.suse.de*) operating system and Apache Web server (*http://www.apache.org*) with the the Tomcat 4.0 Servlet/JSP Container (*http://jakarta.apache.org/tomcat/*).

Computational part of the programs is developed in C and GNU Octave [3]. GNU Octave is a high-level language with syntax compatible to the one of Matlab.

For compilation of C code we have used the standard version of GNU gcc compiler which is part of the Linux distribution.

Gnuplot [2] is used for creation of images.

Client side user interface is developed with J2ME Wireless Toolkit (*http://java.sun.com/j2me/*) from Sun Microsystems (*http://www.sun.com/*).

III. Application description

THE client application can be downloaded from *http://lolek.csc.unist.hr/MathGSM/MathGSM.jad*. MIDP [4] device pointed to this address will automatically follow the link and download necessary files.

A. GSMLab

GNU Octave [3] is a high-level language primarily intended for numerical computations. *GSMLab* is interface for java enabled mobile devices in which user can enter sequence (a program) of Octave commands. After entering the program, computation is started and the commands in the command-input filed are uploaded to server and executed. When computation is done, output of the program is returned to the user. As output can contain both text and images, text output is presented to the user first, without waiting for the graphical part. This way, the response to the user request is faster since the graphical data can be significally larger. If the response contains images, menu is added to the output screen with the list of available images from which user can select one or more images. Selected images are uploaded only on demand, to avoid communication overhead if user is unsatisfied with results.

Although the fact that the user must be well acquainted with the GNU Octave programming language might be considered as a slight disadvantage, the benefits of being able to perform almost any numerical computation make it worth while. Beside, GNU Octave is highly compatible with Matlab programming language which is de facto standard in scientific and engineering calculations.

For users who do not need fully functional programming language, based on this technology specific solutions could be developed for a particular problem. Examples of this approach are programs with web interface developed at Center for Scientific Computing of the University of Split (*http://bolek.csc.unist.hr/czr/en/online.php3*), like simulation of particles settling in a river, on-line computation of highly accurate solutions of eigenvalue problems, etc. These and similar programs can be ported to java enabled mobile devices.

B. GSMPlot

GSMPlot is special module of *GSMLab* application designed for drawing different types of functions. It has easy-to-use interface and graphical output. This module is good example how different types of applications for specific problems can be developed based on this technology.

User can select between five different types of functions:

1. Function of one variable: $y = f(x)$
2. Parametric function: $x = f(t), \quad y = g(t)$
3. Function of one variable in polar coordinates: $r = f(t)$
4. 3D function of two variables: $z = f(x, y)$
5. 3D parametric function of two variables: $x = f(u, v), \quad y = g(u, v), \quad z = h(u, v)$

For each type, the user can set variable ranges, or use default values. Devices that support colors are detected and images are plotted in color which can be helpful to visualize 3D functions (types 4 and 5).

IV. Security issues

As user is provided with fully functional programming language and can write programs which are executed on server side, malicious user could compromise server security. Another problem can arise if the user starts extremely computationally or memory exhausting program. This could occupy all of the server resources thus blocking other services and other users from accessing the server. All of this suggests that security issues of this service are extremely important.

For the reason of the limited resources of the mobile devices which can be used to access this service, security is handled on the server side, and has three levels.

At first level, all of the user input is filtered and in case of the detection of any suspicious command, execution is blocked and the user is informed about the reason. This is basically very primitive method and probably can be avoided by skilled user, but in most cases it is effective enough.

On the second level, application server is started in *chroot-ed* environment, and user can see only the subtree of the main application directory. In this part of the system, limited set of system commands is available, and even with them the user can compromise only subtree of the application. This is standard procedure for lot of Unix services like *file transfer protocol* and others. This protection is good enough to protect system from possible attacks although there are techniques to escape from chroot-jail.

On the third level, server limits system resources that user program can use. Limited resources include maximum execution time, memory limit, number of files opened, size of a file, etc. This limitations can be fine adjusted if necessary.

Although there is no such a system that is absolutely secure, these three levels together provide high level of system security. User is limited with commands that can be used the and resources it can occupy, and locked in a chroot-ed environment. Beside that, additional security can be achieved with user authentication and access logging.

Conclusion

Wide number of new generation consumer electronic devices such as mobile phones and handhelds support java technology and wireless Internet access. Using this new technologies, computationally demanding applications can be brought to end users with access to power of strong servers including parallel computers.

Based on this technology, number of engineering problems could be solved providing fast and accurate numerical solution on-site, without need to take the job to the laboratory.

References

[1] Damir Krstinic and Ivan Slapnicar, *On-line interactive programs for solving and visualization of scientific problems*, MIPRO 2001, Multimedia and Hypermedia Systems

[2] Thomas Williams and Colin Kelley, *Gnuplot, An Interactive Plotting Program*, 1998, `http://www.gnuplot.org`.

[3] John W. Eaton, *GNU Octave*, 1997, `http://www.octave.org`.

[4] Different articles from *http://java.sun.com/j2me/*