Applied Programming on WEB-based Environment

ALEXANDER VAZHENIN, DMITRY VAZHENIN
Computer Software Department
University of Aizu
Tsuruga, Ikki-mach, AizuWakamatsu, Fukushima, 965-8580
JAPAN

Abstract: - The paper describes basic elements of the WWW-oriented Workplace of Applied Programmer (W4AP). From the user's viewpoint, this workplace represents an application-oriented virtual machine being accessible via standard WEB-browsers. This means that the hardware/software features of the W4AP server should be hidden from the applied user. The proposed software has the unified style for different programming platforms, special tools optimizing and controlling the program creation and implementation as well as an environment combining visual and traditional programming technologies.

Key-Words: - WWW-applications, client/server technology, visualization of computations and data, portability, template manipulating subsystem

1 Introduction

The increasing availability of Internet-oriented applications provides a very significant effect on all areas of scientific computations, which can be characterized by the following features:

- A big number of numerical computer experiments.
- A large amount of initial and result data.
- A big and changeable set of powerful computing systems (parallel or sequential) and software tools, which are effective in getting very high performance.
- Complexity of theoretical methods hampering their use by a large number of people.

These characteristics require complementary representation for both data and methods of their processing. This means that the user should have the same style in choosing and/or developing the suitable methods via analysis of initial data as well as features of algorithms. A serious point is how to increase programming productivity of end users. Another serious point is how to reach reusability and portability of programs (including parallel programs).

This confirms also the importance and necessity in researching and developing a simple and unified WWW-oriented system serving remote applications via Internet [1-4]. We think also that an initial point in creating such a system should be the use of a WEB-interface being accessible from any standard browser. This is because such an interface is applicable for all people having a deal with computers. The other issue is that such approach allows hiding specifics of hardware/software features from applied users.

The paper describes basic elements of the WWW-oriented Workplace of Applied Programmer (W4AP). From the user's viewpoint, this workplace represents an application-oriented virtual machine being accessible via standard WEB-browsers. The proposed software has the unified style for different programming platforms, special tools optimizing and controlling the program creation and implementation, comfortable environment combining visual and traditional programming technologies. Section 2 contains an architectural overview of server components as well as demonstrates how the management of application projects is implemented. In section 3, features are shown of the W4AP user's interface and data visualization subsystem.

2 Concepts of W4AP-system

2.1 System Components Overview

One can distinguish the following basic components in a W4AP system (Fig. 1):

- **Client:** provides multiple users interface and submits requests to the W4AP-server.

- **W4AP-server:** receives requests from clients and executes required software modules on their behalf.
- **Local Computing Resource (LCR):** represents a local computer cluster to provide parallel and distributed computations.
- **Remote Computing Resource (RCR):** represents computing systems, which have no direct connections to the Internet, or these connections are very restricted.

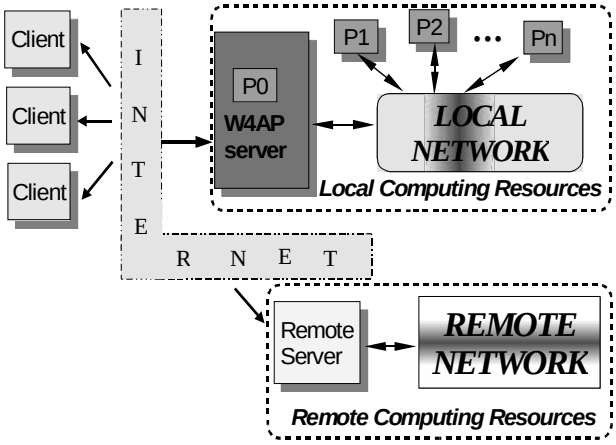


Fig. 1 W4AP: Basic Components

Communications between client and server are implemented via Internet protocols like TCP/IP and HTTP. The interactions between W4AP and LCR-resources are that the W4AP-server is a computational node of this cluster, and implements a part of computations during the problem solution.

Usually, RCR-systems are closed supercomputers and/or internal networks. Therefore, they should have special servers controlling accesses to their resources. An example of such a system can be the network of the University of Aizu based on the Network File System (NFS). In this case, the W4AP-server implements transformations of client requests according to conditions of remote server.

2.2 Software Architecture

Fig. 2 shows the main elements W4AP-server kernel, which controls Internet access, multi-user support as well as security issues. Importantly, **Producer of the source code** implements not only ordinary editing operations on source files. It can also be a visual programming shell build on the server kernel. The user's program implementation is under control of the client/server subsystem storing, manipulating and serving with the user's projects.

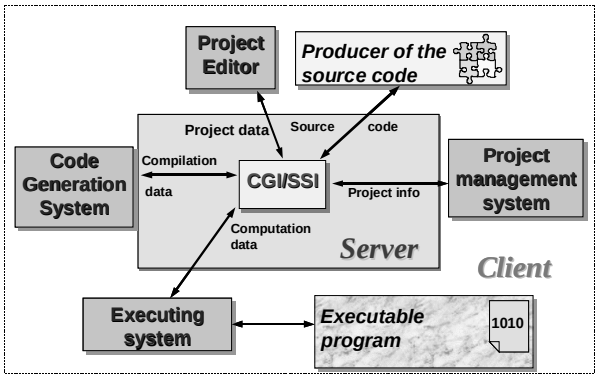


Fig. 2 W4AP: Software Architecture

There are the following types of the user's projects:

- **Sequential project:** is to implement sequential programs on server.
- **Local parallel project:** is to create and execute parallel programs on a computer cluster.
- **Remote parallel project:** is to provide calculations on the Remote Computing Resource.

We think that this flexible organization allows the efficient execution of application programs as well as on-line representation of intermediate and final results.

W4AP-server based on the well-distributed Apache system. This system has rich possibilities in managing remote access. Together with standard HTML-pages, we use also special CGI-scenes as well as the access authorization and resource management. CGI-scenes were developed using the UNIX interpreter Shell. They can also be realized as Perl- or/and PHP-applications. Java-applets are widely used in organizing human/computer dialogue as well controlling and visualizing computations and data. The W4AP system supports different compilers and libraries. It is possible to apply CNU C, Java, etc. Parallel applications are based on Message Passing Interface (MPI). The W4AP server was used to design and developed MPI-programs being implemented on the University of Aizu network.

3 Prototype and implementation

The W4AP-system supports the unified work style for different programming platforms, and includes special tools optimizing and controlling the program creation and implementation.

3.1 W4AP User's Interface

To access the W4AP-system, it is only necessary to have any standard WEB-browser. The work with the system starts after inputting URL. At first, the system checks passwords and invites the user to the main page containing common information as well as the possible choices between projects management, accounts management and maintenance. The project manager allows opening the one from existing projects, creating/removing projects as well as uploading/downloading them from/to client. Each user has only an access to his/her project data. This allows the system safety.

There are the following manipulations with the project:

- Create/Rename/Delete/Edit file.
- Edit makefile and project options.
- Upload and download files.
- Compilation and linking of an application.
- Launch an application.

Fig. 3 shows an example how to implement editing of MPI-project. The user is invited to fill special fields related to the compiler type and options as well as specify the names of computational nodes on which the parallel program will be implemented.

There are special monitors to control errors and program execution. The ERROR-monitor allows checking compilation errors as well as editing a source code. The EXECUTION-monitor shows the trace of program execution. It allows

- Verifying errors during the program implementation,
- Showing results as HTML-document,
- Checking stages of the program implementation ("program is started", "program is in waiting", "program is implementing", "program is finished"),
- Obtaining information about results (report files, visualization of intermediate and result data).

Fig. 4 depicts the EXECUTION-monitor window containing results of the Java LINPACK-test implementation. Fig. 5 shows results of the MPI-project execution using the University of Aizu network. The calculations were implemented on 16 SUN workstations. It is shown the total time of matrix multiplication.

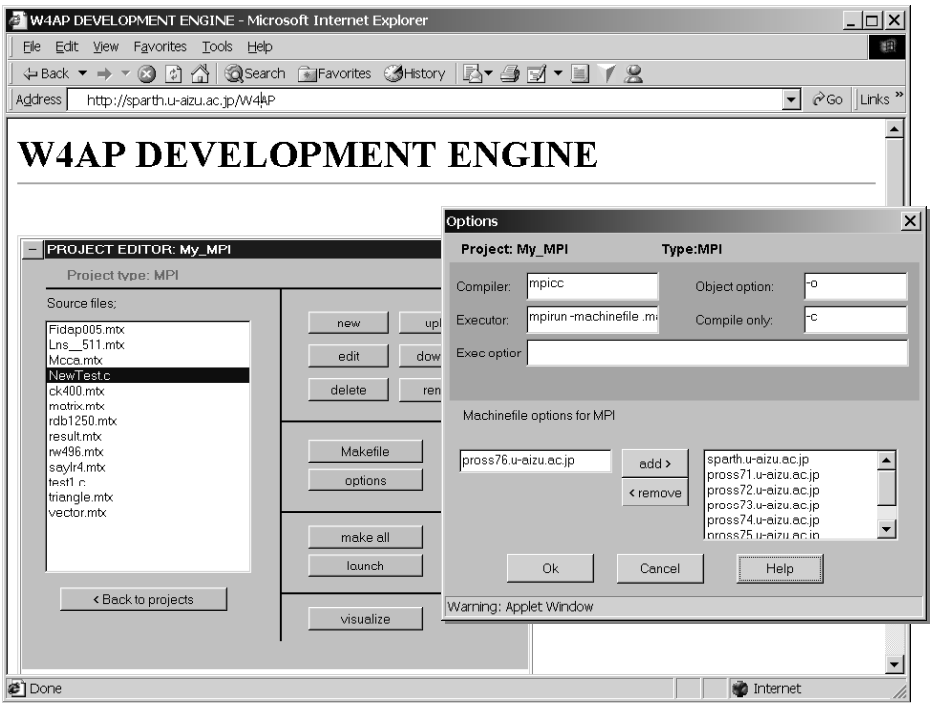


Fig. 3 W4AP: Edit options of MPI-projects

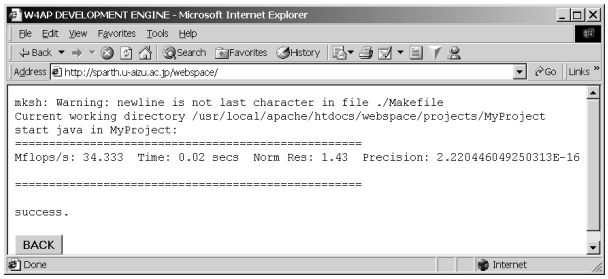


Fig. 4 Implementation of the JAVA-project

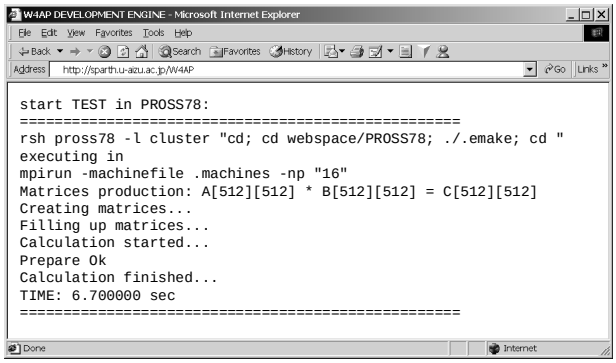


Fig. 5 MPI matrix multiplication

3.2 Package for visual representation of matrix data

The observation of the user's program running can be done by data outputs. To represent initial, intermediate and result data, a special package of functions for the multimedia representation of matrix data has been developed. The detailed description of package functions was done in [5]. The presented software is an applet downloading and implementing on the client computer. Fig. 6 depicts the main window, to specify several functions for the data visualization, and an example of matrix image.

It is possible to implement different functions of matrix filtration, which is to obtain matrices with characteristics allowing their representation on multimedia devices as well as improving the analysis of initial matrix properties. The user has opportunity to choose a function according to *Pattern (Color Scale)*, *Sign* and *Maximal/minimal Values* as well as special functions implementing the verification of *Diagonally dominant* feature. It is also possible to Setup some filtration, visualization and scaling parameters.

In some cases, the size of array should be bounded because of limitation of computer display and/or memory size as well as users' abilities.

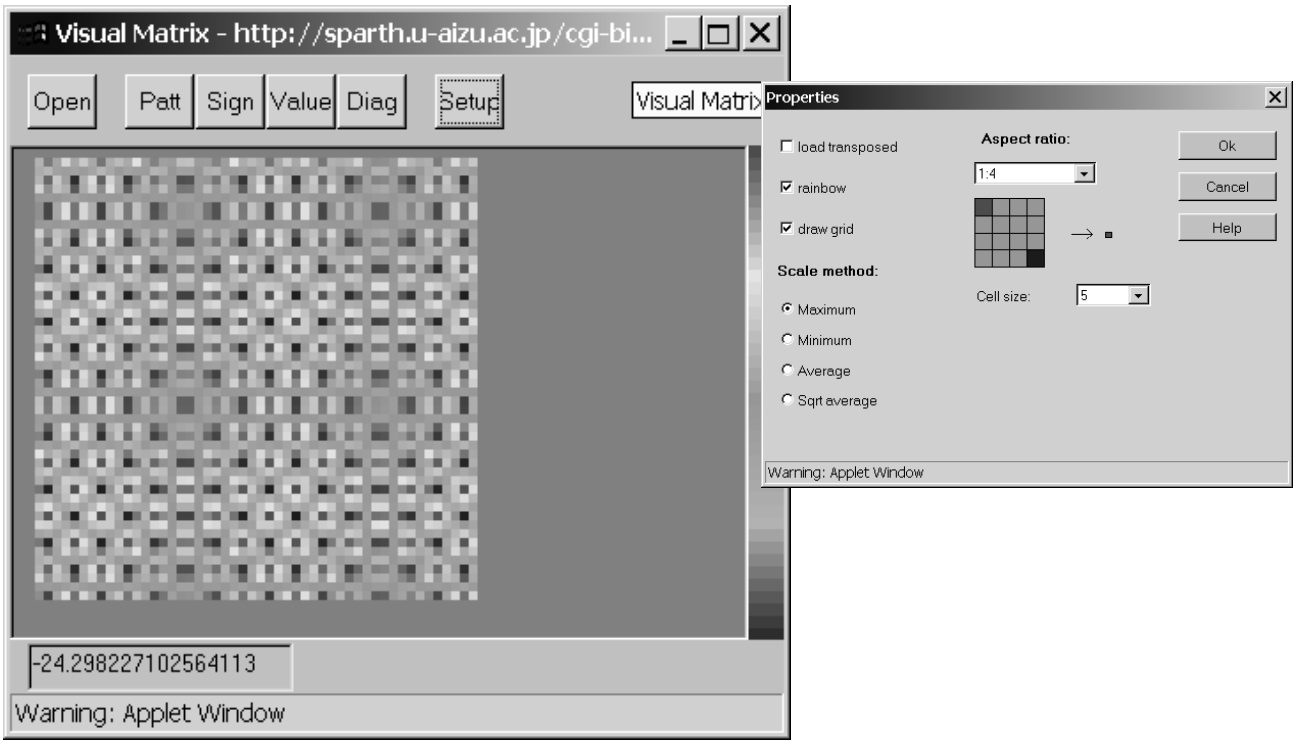


Fig. 6 Visualization of matrix data

Special procedure called *matrix scaling* is to transform an initially huge matrix to a matrix having a suitable size and providing the important features of the initial matrix. Different scaling transformations may surely produce different result matrices emphasizing different features of the initial matrix (Fig. 7).

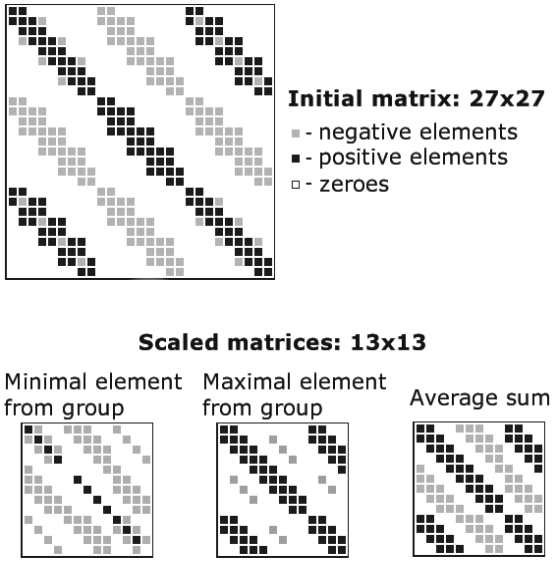
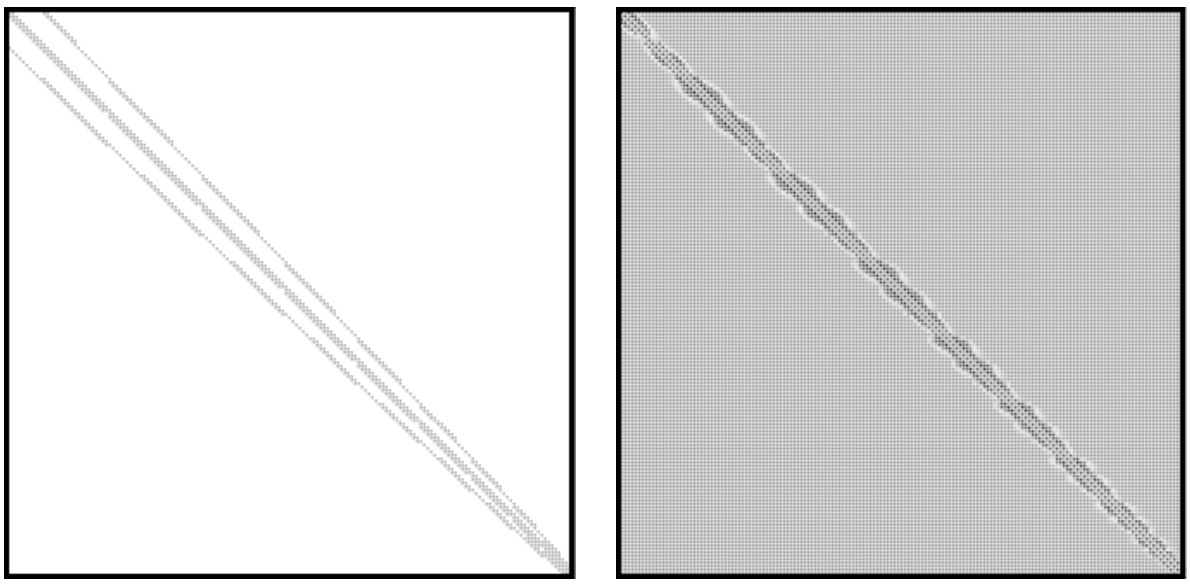


Fig. 7 Matrix Scaling

Special Data Formats are widely used to store/represent matrices of a very large size. Harwell-Boeing Format (HBF), Matrix Market Format, Sparse-Row (SRF) and Sparse-Column (SCF) Formats representation may be examples of such formats, developed not only for adapting numerical packages for concrete applications but also for collecting test matrices.

Fig. 8 contains an example of sparse matrix representation. This matrix was obtained from *http://math.nist.gov/MatrixMarket/data/NEP/*. It is stored in Matrix Market Format, and has the size of 496x496. We have scaled it down in three times by selection of Maximum. This strategy saves main features of initial matrix especially the absence of the diagonal dominance.

It is important to note that the scaling procedure decreases details, and that is why the result matrix may be greatly different from the initial arrays. In this case, the combination of several functions allows implementing a flexible work with large numerical arrays and highlight necessary information depending on user's opinion.



a). The sign portray

b). Absence of the Diagonal Dominance

Fig. 8 Representation of sparse matrix

4 Conclusion

We have presented main architectural elements and features of human/computer interactions with the W4AP-server. The use of standard WWW-interface allows to hide hardware specifics and to reduce working conditions to the standard HTML-environment. The software presented is installed in the University of Aizu and used in student research projects. The practical experiments with the system confirm its good adaptability to the user's profile because of the same environment for different programming platforms.

Our future work is to extend the W4AP-platform by modern data visualization systems like QT, VTK, as well as the application-oriented visual programming technology based on presentation of application methods in the film format [6,7]. Within the framework of this technology, self-explained series of stills presenting computational schemes are used. Each scheme reflects some knowledge about data processing. It defines sets of points and/or moving objects in multi-dimensional space-time, and possibly a partial order of scanning of these points and objects. Each film can be considered as a 4D space-time object or component. The user embeds his/her algorithmic ideas into a computational scheme (by making this scheme more precise) or into a composition of such schemes.

We expect that the combination of W4AP-approach with self-explanatory films and multimedia format for data representation will be a very attractive basis for distance learning materials as well as for knowledge-on-demand applications.

References:

[1] T. Berners-Lee, WWW: Past, Present, and Future, *Computer*, Vol. 29, N 10, 1998, pp. 69-77.

[2] H. Casanova, S. Matsuoka, J. Dongarra, Network-Enabled Server Systems: Deploying Scientific Simulations on the Grid, *Proc. of the 2001 High Performance Computing Symposium (HPC'01)*, Seattle, Washington, USA, 2001, pp. 83-91.

[3] M. Singh, The Service Web, *Internet Computing*, Vol.4, N 4, 2000, pp. 4-5.

[4] R.-D. Schimkat, S. Mueller, W. Kuechlin, R. Krautter, A. Lightweight, Message-oriented Application Server for the WWW. *Proc. of the 15th ACM Symposium on Applied Computing (SAC 2000), Como, Italy*, ACM Press, 2000, pp. 934-941.

[5] A. Vazhenin, N. Mirenkov, D. Vazhenin, Multimedia representation of matrix computations and data, *Proc. of the Fifth Joint Conference on Information Sciences*, Atlantic City, USA, 2000, pp. 592-595.

[6] N. Mirenkov, A. Vazhenin, R. Yoshioka, T. Ebihara, T. Hirotomi, T. Mirenkova, Self-explanatory Components: a New Programming Paradigm, *International Journal of Software Engineering and Knowledge Engineering*, Vol.11, N 1, 2001, pp. 5-36.

[7] A. Vazhenin, N. Mirenkov, Visual Programming System VIM, *Programming and Computer Software*, Vol. 27, N 4, 2001, pp. 217-231.