

**Meta $\Psi$ : A Web-based Metacomputing Problem-Solving Environment for building Complex Applications**

by Ranieri Baraglia and Domenico Laforenza

The increasing complexity of large distributed scientific applications raises the problem of the coordination of diverse computational resources (computers, data bases, etc.). Multi-disciplinary applications often make use of coupled computational resources that cannot be replicated at a single site. There is the need for smart and user-friendly Problem-Solving Environments (PSE) that free scientists from concerns related to the location and complexity of the computing platform being used.

In this article we describe the main features of Meta $\Psi$ , a software tool developed at CNUCE-CNR to build PSEs for the execution of complex applications on a Web-based metacomputer. This tool is designed to supply a completely transparent support to the user, who thus does not need to be aware of the location and the allocation of computing resources.

Meta $\Psi$  has a 3ier architecture with the following layers, as shown in Figure 1:

- Client side: a Web browser (eg Netscape Navigator)
- Middleware: Web Server and LDAP (Lightweight Directory Access Protocol) Server
- Back end: the set of computing resources.

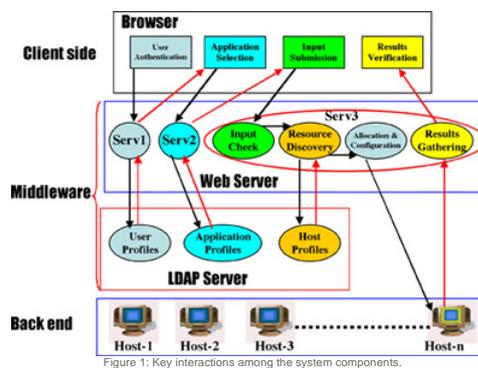


Figure 1: Key interactions among the system components.

Meta $\Psi$  has been implemented on an appropriately extended web server, mainly using standard tools. In particular, the Java Servlet and Directory Service facilities of LDAP have been used. As can be seen, in order to guarantee easy maintenance and extensibility of the prototype, the design is modular.

The client side consists of a Web browser representing the graphical interface which guides the user when selecting an application, and in the input of data and the retrieval of results. After authorisation (see Figure 2b), the user selects the application (Figure 3a) and then provides the input data (Figure 3b).



Figure 2a (top) and 2b (bottom): Welcome and User Authentication Windows.

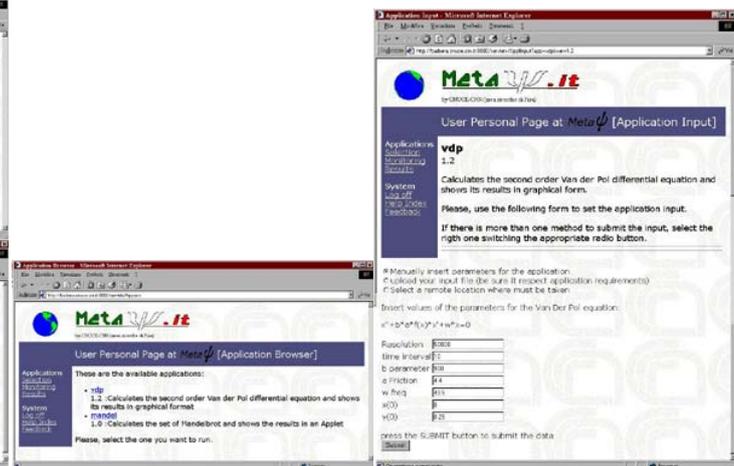


Figure 3a (left-hand side) and 3b (right-hand side): Application Selection and Input Windows.

The middleware (see Figure 1) layer consists of a Web server and an LDAP server. The Web server takes care of the interaction with the client and executes the Java servlets handling the user requests. The servlets residing on the Web server are: Serv1 which handles the user authentication stage; user profiles establish who can do what and where results of a run should be stored. Serv2 handles the application profiles management, and guides the user in the data input, according to the selected application. Serv3 is the most complex servlet taking care of validation of input data, resource location, allocation and configuration, remote execution of the application, collection and forwarding of results. When responding to user requests, the middleware employs LDAP functions to locate those computing resources that can provide the requested services. LDAP provides information on the computing resources by accessing a Directory Information Tree (DIT). The tree consists of entries representing the computing resources through a group of attributes. The server then activates a remote execution of the selected application. HTML forms are used in the interaction between the user and the Web server. These forms activate the Java servlet that executes the requested action (application selection, data input, etc.). The initial page (see Figure 2b) allows the specification of the userid and password. In order to control access to the system's resources, different user profiles can be defined according to a predefined strategy, implemented using LDAP. After authentication, the user is offered a list of applications that can be run on the machines belonging to the system. An application is selected by clicking on the hyperlink related

to the application. Each application has an associated profile describing its computational characteristics/requirements: eg, name, version, documentation available, type of input required, sources of data, etc. The application profile, stored in the LDAP server, is exploited by the Web server to drive the input process. An HTML page guiding the data input according to the characteristics of an application is produced (see Figure 3a). The input data can be submitted (see Figure 3b) to the application in three different ways: by manual data entry, by selecting a file from the client's local disk, or by choosing a link to a remote data source (ie, a file located on a metacomputer).

The back end (see Figure 1) is made up by high performing computing resources, multiprocessor systems and workstation networks which provide computing power to the applications of the metacomputer. The Web server has user accounts on these machines that allow the execution of the applications. In general, the execution of an application on the back end takes a significant amount of time; the user can thus exit the metacomputing session once the application has been submitted. At job completion, the user is notified by an email message where to access the page of the results built by a Java servlet. When execution is completed, the results are passed to the server that forwards them to the Client (Figure 4).

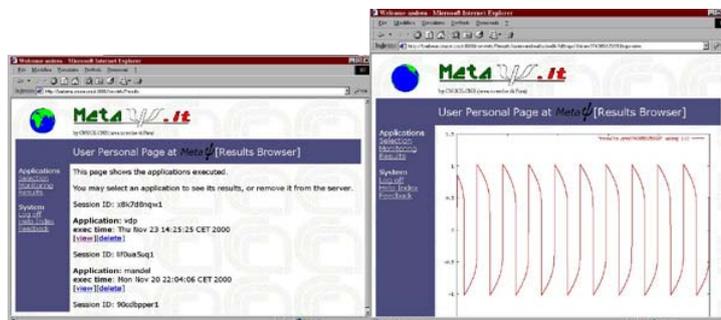


Figure 4a (left-hand side) and 4b (right-hand side): Application Execution Control and Results Verification Windows.

In the near future Meta $\Psi$  will be used to implement SIMBEX, a simulation of crossed molecular beam experiments. SIMBEX is a computational procedure based on a priori calculations of structures and processes of molecular systems. The procedure runs over several modules; each module consists of alternative or coordinated computer codes which accomplish particular tasks.

**Please contact:**

Ranieri Baraglia, Domenico Laforenza - CNUCE-CNR

E-mail: {[ranieri.baraglia](mailto:ranieri.baraglia@cnuce.cnr.it), [domenico.laforenza](mailto:domenico.laforenza@cnuce.cnr.it)}@cnuce.cnr.it