

Accessing NQS via WEB: a simple user interface.

Sergio Bernardi, CINECA, via Magnanelli 6/3, I-40033
Casalecchio di Reno, Italy, s.bernardi@ceneca.it

ABSTRACT: *A simple HTML/Java user interface has been written to provide easier access to the NQS queues of our C90/T3D system. It is aimed to those users who do not have any access to NQE client systems either because they have a non Unix PC based workstation or because of their activity find themselves accessing the CINECA facility from many different locations. The CERN http daemon has been installed on the C90 and a set of HTML and cgi-bin scripts and programs has been written in order to provide controlled access to the system. The OPIE One Time Password package is used to provide a more secure access to the user resources. The basic OTP interface is provided by Java applet or by local OTP calculator, depending on the user needs and preferences.*

Introduction.

By the end of 1994 CINECA decided to abandon the VMS and VM Station software and adopt a distributed solution to provide NQS batch access to its C90 system. We carried out some tests with RQS and Superlink and though the results were not completely satisfactory, we planned a migration path to move away from the Station environment part of our user community.

In May 1995 we installed the first release of NQE and because of budget considerations we chose to purchase a master server license only for our SGI CHALLENGE L system and to promote the use of Superlink and the NQE clients.

This was a major shift for many of our users, who had to accept some obscure but peculiar aspect of NQE, last but not least the command user interface.

The starting point.

Since its first release, NQE has offered the opportunity to build an effective Web user interface to get access to batch services. In order to evaluate such an opportunity, some preliminary experiments have been carried out using a set of NQE clients installed in our LAN.

The original package implements a Web interface that relies completely on the NQE client component.

The main characteristics of this implementation are:

- the Web server runs on the NQE client system
- standard NQE user validation
- shared job log output directory

The basic interface features available are:

- script submission

- script edit and submission
- request status display
- request delete and signaling
- job output list and retrieval

Unfortunately, this NQE client based approach does not reflect the environment that our user community would appreciate most. Since the very beginning we discovered that NQE was not exactly as successful as expected. Who was used to the centralized Station service, found that telnet was a more comfortable solution to batch access needs, without having to install a piece of software that in the long term might mean maintenance troubles and necessary upgrades.

Moreover, there are other environment related aspects that play an important role:

- a considerable amount of our users have only MS Windows based workstations
- they may not have access to a system running the NQE client software
- CINECA does not provide access to its internally running NQE clients

We then decided to introduce some modifications to the model proposed by the original NQE Web example:

- no need to access a NQE client
- user validation and access security provided by the interface
- a different interface look

The Web based interface components.

The components of our Web based interface are essentially two:

- access control and user validation

- NQS command interface

These components are implemented by HTML pages, shell scripts, Java applets and C programs. A Web server based on the CERN http daemon was therefore installed on our C90 Unicos 9.0 system, in order to provide the necessary basic environment.

The access control and user validation.

To access the interface, users must validate their Unicos username. Although possible, we do not use the standard user/password protection scheme provided by http itself. The choice is based on simple reasons:

- user passwords traffic through an intrinsic insecure network should be avoided as much as possible
- the user of the interface does not retrieve HTML pages, but gets access to a service instead

The solution to the user validation problem came from the OPIE (One-time Password In Everything) package from NRL, that we ported to Unicos two years ago.

The OPIE software provides a one-time password system that can be used to reduce and eventually eliminate password traffic.

We follow a two steps approach to validate users:

- the username must exist in the system UDB
- the user must give OPIE a valid one-time password

Once the above steps are successfully completed, the HTML page containing the main operation menu is delivered to the requesting Web browser.

The access and user validation component provides the essential HTML pages to perform the following functions:

- enter the user login name at access time (fig. 1)
- enter the personal secret string for one-time password calculation (fig. 2)
- register or reset the personal OPIE entry when necessary (fig. 3)

A basic component of the OPIE package is the one-time password calculator. The program that implements the calculation algorithm is available on many different platforms as well as under the form of Java class library.

If the user does not have a local version of the OPIE calculator then that user can take advantage of the interface Java applet that makes available the necessary functionality.

Since the Web server context is essentially a stateless environment, some information has to be kept in order to guarantee that after the validation every incoming request is legal.

In order to build a safe environment the validation process stores information to identify the user and associates a token to the virtual session created. Such a token is from now on passed back and forth between the client browser and the server scripts to identify a particular session.

The NQS command interface.

The second component includes all the HTML pages, programs and shell scripts that makes available the NQS commands to the user. Every request is processed by a common script that dispatches the task associated to the appropriate program or shell script.

The NQS command is then executed after switching to the requesting user.

The user selects from a choice menu that give essentially the following basic features:

- job script submission
- job script edit and submission (cut and paste provided by the basic browser functionality)
- request status
- MPP request status
- request delete
- request job output list and file retrieve

The interface look takes advantage of the HTML frame enhancement that the most common browsers support and that seems to be very popular nowadays.

The figure 4 shows the frame layout.

The Web based interface has been tested with three of the major browsers:

- Netscape
- MS Internet Explorer
- Mosaic

For those who do not like frames or use a browser that does not support this enhancement, there is a frameless version too.

The Web for accessing services.

There is no doubt that the Web has a tremendous potential as means for providing access to host based services like NQS for instance. There are many examples of complex Web based applications that found their way to a large number of users. The effort to develop a Web server for providing service access is not so compelling, and give the user access from any point in the network from virtually any type of platform.

As far as our specific Web server development is concerned, we focused our attention on the following aspects:

- the cgi-bin scripts and programs are the core of the service
- the Web server has to have some protection against unauthorized access
- the interface appearance, though important, should be simple and clean for a fast display time
- the use of fancy/exotic HTML features should be avoided, because most of the time they are not widely supported

Conclusion.

At present the Web server and the access interface are not user available. Some more specific tests of the user validation process are under way and at the same time we would like to extend the same approach to the T3E system we recently installed.

There are plans to add new features like:

- local job script submit via Java applet
- automatic refresh of the request status display
- execution server NQS queues and CPU load display

References

- [1] Designing large-scale Web sites - a visual design methodology, Darrell Sano, Wiley Computer Publishing, 1996.
- [2] Exploring JAVA, Patrick Niemeyer and Joshua Peck. O'Really & Associates, 1996.

Web links

- [1] [/www.cs.umd.edu/~harry/jotp/](http://www.cs.umd.edu/~harry/jotp/)
- [2] [/ftp.nrl.navy.mil/pub/security/opie](ftp://ftp.nrl.navy.mil/pub/security/opie)

Appendix



figure 1

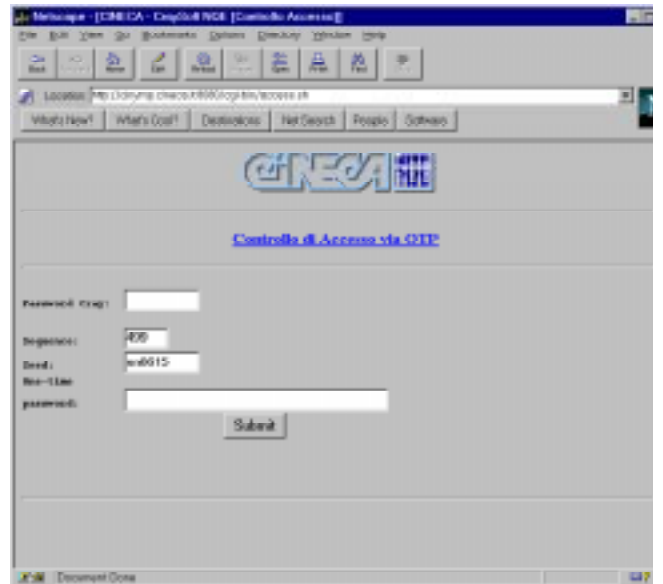


figure 3

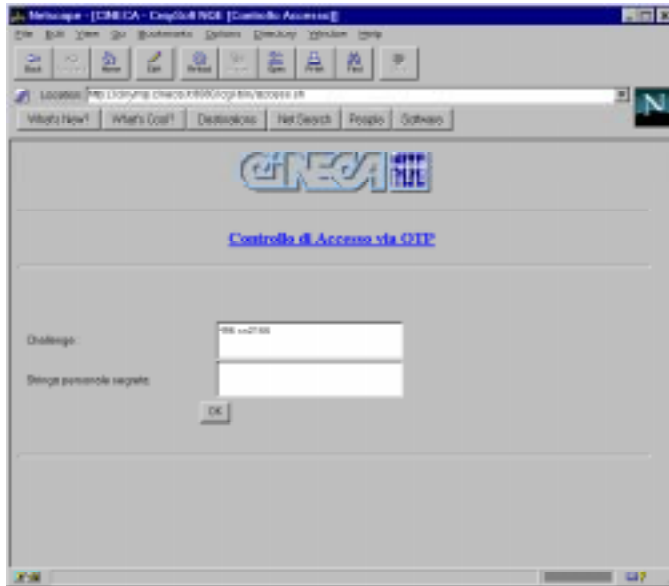


figure 2



figure 4