Fast Access to Remote Objects 2.0 A renewed gateway to ENEAGRID distributed computing resources

Angelo Mariano*, Giulio D'Amato†, Fiorenzo Ambrosino‡, Giuseppe Aprea§, Antonio Colavincenzo‡, Marco Fina†, Agostino Funel‡, Guido Guarnieri‡, Filippo Palombi§, Samuele Pierattini¶, Giovanni Ponti‡, Giuseppe Santomauro‖, Giovanni Bracco§ and Silvio Migliori§

Energy Technologies Department, ICT Division
ENEA
*Bari, ‡Portici (NA), §Rome, ¶Florence, Italy
*Email: angelo.mariano@enea.it

†Sysman Progetti & Servizi srl
Rome, Italy

‖Consortium GARR
Rome, Italy

***Abstract*—This paper introduces a renewed gateway to ENEAGRID distributed computing resources, named Fast Access to Remote Objects 2.0 (FARO 2.0). FARO 2.0 is a tool for application and desktop virtualization with a strong focus towards user experience (UX), providing trained as well as untrained users a collection of centralized services that can be seamlessly used on their client through a remote desktop protocol. FARO 2.0 is a JavaFX application whose graphical user interface (GUI) and whose main logics has been implemented through the well-known Web technologies (HTML5, CSS3, Javascript) for a easier maintainability and customizability, taking full advantage of the WebView component. Its framework has been deployed both as general purpose GUI for remote user access to ENEAGRID resources and as specialized application or workflow oriented GUI. They are applied in a set of applicative domains, ranging from material science to technologies for energy and industry, environmental modeling and nuclear fusion. Some examples and results are also presented.**

***Keywords*—graphical user interface; remote desktop; distributed computing; virtual labs**

I. Introduction

The need for a centralized access to computing resources has resulted in an extensive research towards software tools, such as scientific gateways, to allow an intuitive and easy-to-use fruition of remote services (e.g. application and desktop virtualization, storage, data ...). Distributed computing infrastructures (such as grids, clouds), as well as general-purpose yet highly specialized cloud facilities, are now empowering scientists in a growing number of tasks from their everyday work life. Furthermore, they are even becoming a key foundation technology even for untrained users in many areas beyond scientific research, such as in education. For this reason, modern scientific gateways should be developed with a strong focus towards user experience (UX), through a high degree of interactivity in a pleasing graphical user interface (GUI), just like any mainstream application for smartphone and tablets (an "app") that users already understand and appreciate. Moreover, they should foresee a high degree of maintenability and customizability, through the usage of well-known technologies. ENEA, the Italian National Agency for New Technologies, Energy and Sustainable Economic Development is offering a completely renewed gateway for its High Performance Computing (HPC) services, called FARO 2.0 (Fast Access to Remote Objects) to let internal as well as external scientists take advantage of its remote services based on ENEAGRID distributed computing resources and CRESCO linux clusters [1].

II. Main picture

Developed in the contest of the Italian PON "Smart Cities and Communities" R&C 2007-2013 with the project "EDOC@Work 3.0 - Education and work in the cloud" [2] in collaboration with Sysman Progetti & Servizi S.r.l. [3], FARO 2.0 is a tool for scientists and students to perform research as well as train in a real-world HPC environment. FARO 2.0 has been implemented as a drop-in replacement for its predecessor, FARO [4], it thus relies on its same network infrastructure, depicted in Fig.1, and remote desktop protocol. The main goal of FARO 2.0 is to provide users a software solution with a strong focus towards UX, maintainability and customizability, being a pleasant interface for real-time application and desktop virtualization (with remote graphics acceleration) on conventional as well as on "special" hardware platforms (featuring Intel MICs or high performance GPUs).

The remote desktop protocol used to deliver applications and desktops to end-users is based on NX over an SSH. The compression and transport protocol of NX is used to enhance the native X display protocol performances in such a way that services are usable even with slower links. The NX procotol has been released by Nomachine [5], but many open source server and client implementations exists [6], [7]. FARO 2.0 remote desktop server is based on a customized version of FreeNX, where we implemented some advanced features (e.g. load balancing and session distribution over a cluster).

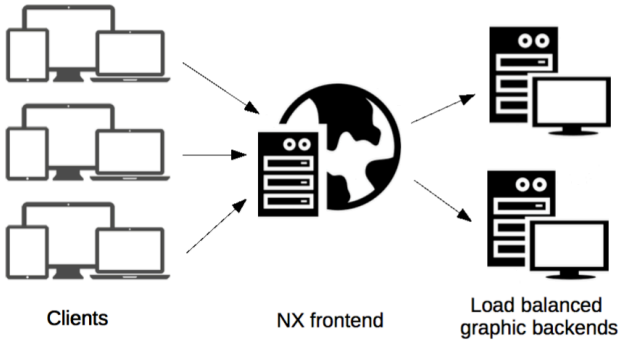

Fig. 1. Network infrastructure: many client devices connect to NX frontend through a NX over SSH protocol and then the frontend redirects and load balances requests over a set of graphic backends that displays the FARO 2.0 interface.

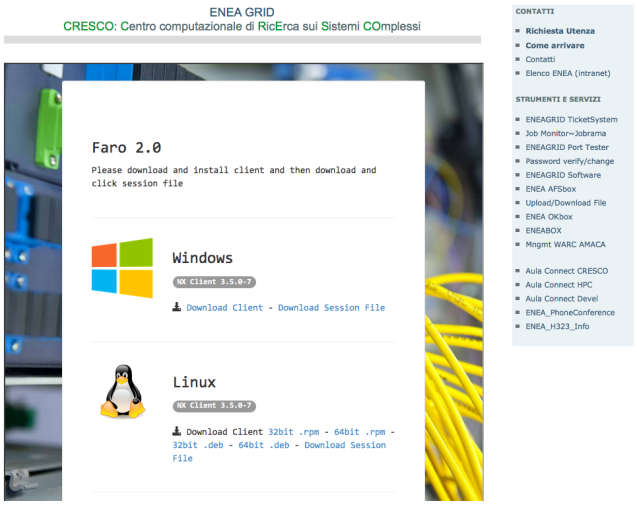

Fig. 2. Web page showing download option for pre-configured NX client.

Accreditated users can connect to FARO 2.0 by downloading a pre-configured NX client (as well as a session file) from its web portal (see Fig.2).

It seems important to remark that application and desktop virtualization is one among the many tools and services that ENEAGRID offers to its users and administrators, such as web portals for batch job submission and monitoring (JobRama), cloud storage (AFSBox), distributed file system monitoring (AMACA) and help-desk (GridTicket). The Authentication and Authorization Infrastructure (AAI) of ENEAGRID is based on Kerberos 5 [8]; it is integrated with AFS distributed filesystem and its Access Control List (ACL) system [9] and is used for all the services provided by the infrastructure (see Fig.3).

III. FARO 2.0

FARO 2.0 is a JavaFX application whose graphical user interface (GUI) and whose main logic has been implemented through well-known Web technologies (HTML5, CSS3, Javascript) for an easier maintainability and customizability, taking full advantage of the WebView component.

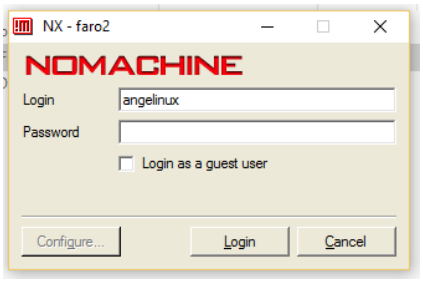

Fig. 3. NX client asking for ENEAGRID credentials based on Kerberos 5 authentication protocol.

The application is delivered to the end-user (as soon as his NX session is authenticated) by a load balanced graphic backend in a CRESCO cluster: this means that FARO 2.0 is executed on ENEAGRID and is served through an entirely managed ecosystem, where a team of administrators takes care of the runtime environment in order to ensure its consistent behavior. Consider that all software packages and components, including the Java runtime environment required by JavaFX, are provided and deployed on a distributed filesystem that connects all computing resources in ENEAGRID. Thus, the only component that users have to install on their machines is the NX client, whereas any other tool and service is remotely executed and rendered on the local screen with near native performances through the NX protocol. This tool and service distribution model ensures better security for data and simpler deployment of complex scientific softwares and their updates, since everything runs on machines that are monitored and serviced by our administrators.

The core of FARO 2.0 is its desktop application (a container, in this context) developed in JavaFX, showing a WebView. Instead, its GUI and its main logics have been developed in HTML5, CSS3 and Javascript as a single page application (SPA), replicating the trending paradigm of cross-platform hybrid apps. The container is a lightweight piece of software that implements an ordinary browser, rendering a local web page. However, the container also injects a custom Java class (a bridge) in the Javascript global namespace. Thus, through Javascript calls, the above web page is able to execute any member function that has been made available from the container. In this implementation, the bridge exports methods to launch new processes based on CLI commands and implements the application logic needed to callback registered Javascript member functions in case of a new message in the standard-out or in the standard-error streams; this allows to virtually execute any command on the host backend server that is executing FARO 2.0. Moreover, every command executed is tracked in a user home log that allows administrators to troubleshoot every problem may occur in the execution of remote code. A convenient call from the Javascript runtime environment is, for example, able to let the user remotely

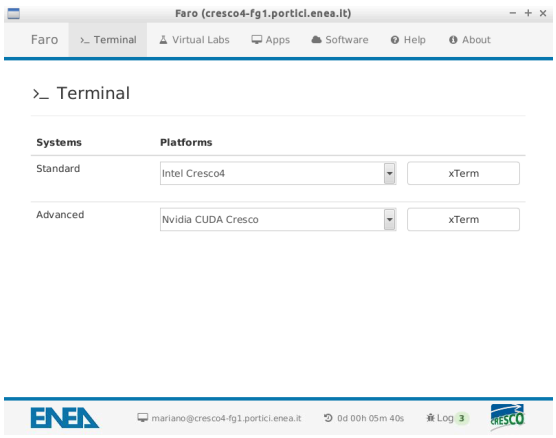

Fig. 4. Main FARO 2.0 interface, showing xTerm launchers for clusters available in the ENEAGRID environment

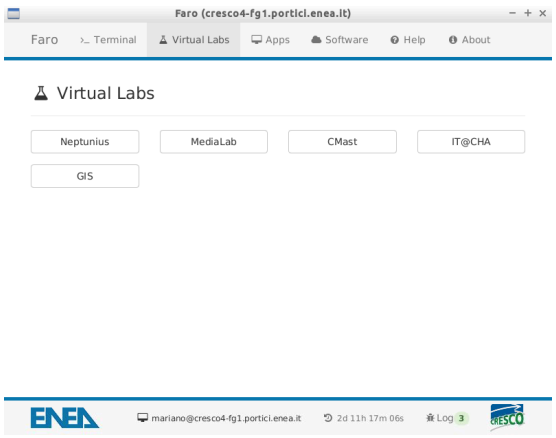

Fig. 5. Main FARO 2.0 interface, showing some virtual labs available in the ENEAGRID environment

open and interact with the command prompt of a standard or an advanced node in a cluster, or launch any kind of scientific software in a dedicated environment (e.g. MATLAB, COMSOL, IDL). More precisely, each launch initiated from FARO 2.0 is redirected through the SSH protocol to the ENEA instance of IBM Load Sharing Facility (LSF), the scheduler that enqueues requests both for interactive and batch jobs [10]. As a matter of fact, any interaction with the ENEAGRID environment can be routed through FARO 2.0. The main interface is shown in Fig.4-5-6.

FARO 2.0 is easy to maintain and customize, since most of its code reside outside the container package and merely implements a SPA (with bindings to some Java member functions): this allows to reconfigure quite the entire application without the need for a compile process. Moreover, the SPA can run (with a limited functionality) within a web browser; it can thus be previewed in a very agile way. We are even able to provide deeply customized interfaces by creating branches of the SPA, without the need to modify the container. There is definitely no need to know how FARO 2.0 actually manages the launch of remote tools and services in order to create

Fig. 6. Main FARO 2.0 interface, showing software available in the ENEAGRID environment

new virtual research environments (e.g. gateways for highly specialized use cases, collecting scientific tools targeting a well-defined research context), such as what we call the "ENEA Virtual labs" (as an example see [14]). This science gateway plays an important role also on computational chemistry applications based on software like Quantum Espresso and cp2k, remotely managed by ENEAGRID CMAST Virtual Lab [15]. From a security standpoint, it seems important to remark that any CLI command executed by FARO 2.0 is authenticated as if the user wrote the command on a shell, and there is no way to execute a malicious command with more privileges (e.g. privilege escalation) from FARO 2.0 code. Moreover, the WebView scope is constrained to local pages in order to avoid cross-site scripting.

Through the usage of web standards, FARO 2.0 implements a full-featured gateway for application and desktop virtualization in the ENEAGRID distributed computing environment that can also be used as a boilerplate for new applications, such as the "ENEA Virtual Labs".

IV. Comparison with other tools

FARO 2.0 offers many similarities with a commercial tool named RemoteApp from Microsoft Corp [11]. RemoteApp is a software tool based on the Microsoft Remote Desktop Protocol implementing a launcher for applications hosted on Microsoft Azure. It must be installed on the user device and is available for Windows, iOS, Mac OS X and Android. This solution is not as flexible as FARO 2.0 because it requires a set of specific OS and resources in order to deliver a service remotely. Another platform that we considered is Citrix Workspace Cloud, that is built around Citrix proprietary protocol [12]. It offers a complete solution to every remote workspace request. This protocol was used in the past by ENEA [13] but was abandoned due to commercial costs and more stringent technical requirements. Instead, FARO 2.0 is distributed toward users as a remote application itself, and for this reason it works on any platform where an NX client is available. Besides this, consider that the JavaFX desktop application at the core of

FARO 2.0 is in some way independent from remote desktop protocol, as it can be deployed through every protocol one can choose to render a remote desktop. The NX technology has been GPL-licensed until the 3.5 version and until now it offers a set of open source and free tools that can ensure a lot of flexibility for our solution. Its protocol is optimized for low bandwidth connections and its performances attracted interests even from Google [16]. We successfully employed this technology in ENEAGRID even for heavy tasks like remote 3D rendering [17].

V. Applications

In the following we will show some interesting applications and customization of FARO 2.0.

The first application is related to the ADP Virtual Lab [18]. The aim is to provide a launcher for a code for the analysis of small-powered wood biomass energy systems (so called COGEGNO). The code has been implemented in Matlab and Scilab environments, and transferred on a web platform, in a "virtual laboratory" powered by the computational systems of ENEAGRID. Among the main components of the model there are the gas recirculation and air staging mobile-grid burner and the thermic gain obtained from the pre-heating of the air in input to the boiler. For the model of the boiler the code uses an helicoidal flow one, with particular attention to the molten salts, for the advantages arising from the possibility offered by these fluids to work in a broader temperature range and at higher temperatures, increasing the gain of the plant. If the power and the electrical efficiency of the cogeneration system are known, the code is able to calculate the performance and the sizes of the main components of the heat generator not only in nominal conditions but also if the operational conditions vary. The code is installed on the AFS filesystem. In particular Scilab code was compiled and only the executable files are stored in AFS. Using the FARO 2.0 interface, users can pass the values of the parameters to the code and choose the operational conditions between dynamically loaded options. These parameters are passed to CLI commands through an easy-to-understand JSON format, that the backend computational core is able to interpret and then execute. A sample illustration of the customized interface is in Fig.7-8-9. Consider that the entire architecture design of FARO 2.0 has strict security policies based on AFS ACLs and allow software developers to deploy their products hiding the implementation; in this way any user can execute scientific codes without accessing directly to them.

A second application of FARO 2.0 is developed for the Web Crawling Virtual Lab [19]. The aim of this tool integrated in the related ENEA project is to create a simple interface beetwen the users and the Web crawling environment installed on ENEAGRID. The Web crawling is an activity that automatically and systematically explores the Web, in order to search for contents/documents to download. Starting from the main html page of the interface, an user can create a web crawling session, launch the session and monitor the generated internet traffic. More specifically, this interface is composed of

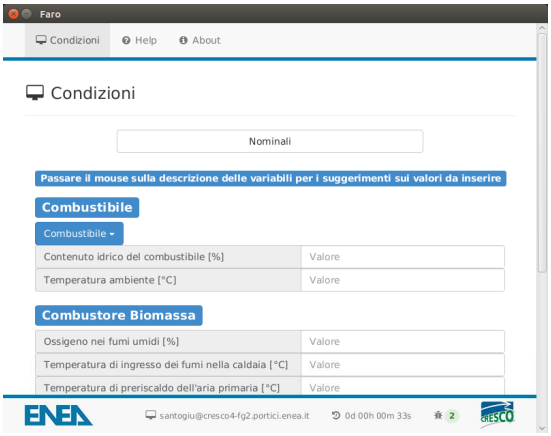

Fig. 7. Customized FARO 2.0 interface for the ADP project

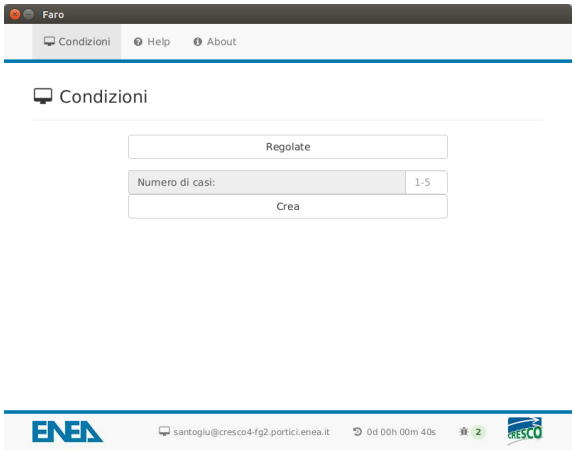

Fig. 8. Customized FARO 2.0 interface for the ADP project: dynamically changing interface (A)

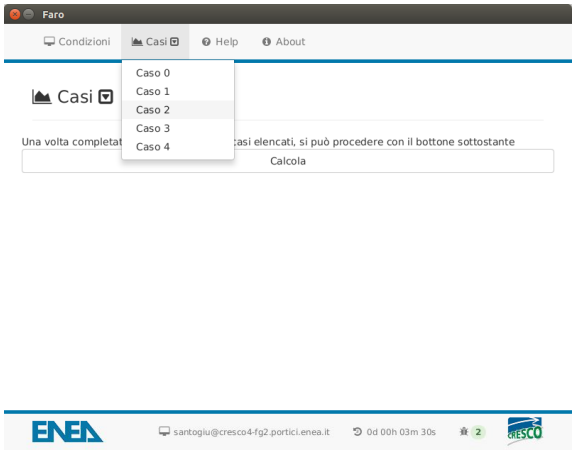

Fig. 9. Customized FARO 2.0 interface for the ADP project: dynamically changing interface (B)

five tabs, in each of them, one can perform some operations. In the first tab, there is a form where one can insert some general informations (such as the title and comments) about the session that wants to create. In the second tab, a user

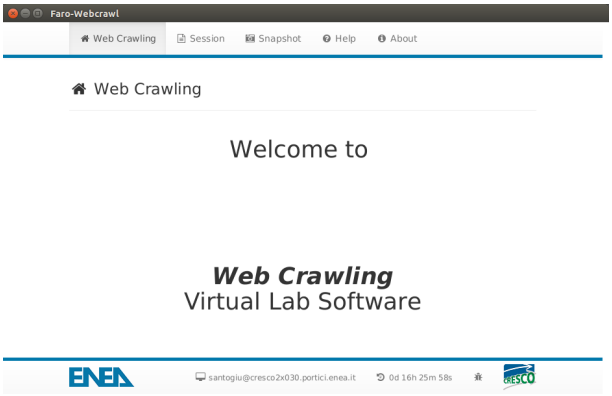

Fig. 10. Customized FARO 2.0 interface: welcome screen for web crawling project

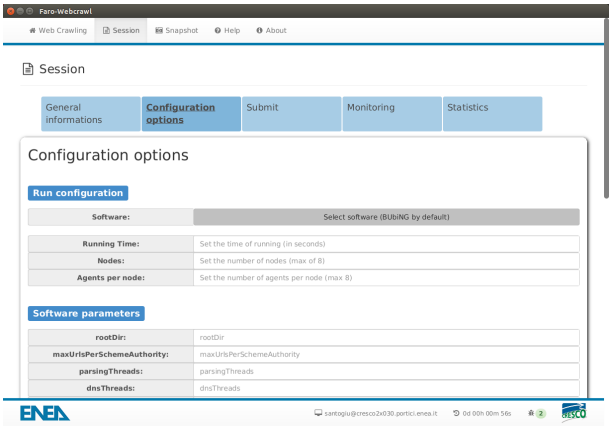

Fig. 11. Customized FARO 2.0 interface: configuration panel for web crawling project

Fig. 12. Experimental FARO 2.0 setup on a Cendio ThinLinc protocol: login form integrated with ENEAGRID Kerberos 5 domain

can fill in a module with the configuration options to pass to resource scheduler in order to submit a session job. These options are split between the run configurations and software parameters. As run configuration an user can set the running time of job, the number of nodes to use and the number of agents per node. An agent is a Java process that plays the role of a crawler (i.e. a software that explores and downloads the web pages). As software parameters it is possible to can set many options for each crawler (e.g., the URLs seed to initially consider, the number of threads and the size of cache memories). If the user does not decide to set these options, automatically the interface creates a session configuration file with default options. In the third tab, an user can submit a batch job for the current session. In the fourth tab, once a web crawling session is submitted, the user can monitoring in real time the downloaded data amount by every agent. Finally, the fifth tab allows to view some statistics at the end of the latest executed session. The tool has been recently equipped with a further feature called "snapshot", which allows to schedule periodic web crawling sessions. This is particularly useful to create a set of snapshots for some portions of the web, in order to analyze the changes and the evolution over a time period. Periodic scheduled web crawling session parameters can be set in the proper section in the tool interface. Customized interfaces are shown in Fig.10-11

For both applications, in the submission step, ad hoc shell scripts are used. These scripts read the parameters defined on the mask and usually encoded in a JSON format and launch the codes on execution nodes following the rules defined by the LSF scheduler. The applications provide also an Help page with an useful set of links for users (to submit tickets to ENEAGRID administrators and/or to monitor job execution).

Every customized FARO 2.0 application is a workflow-oriented GUI enabling interaction with ENEAGRID computational resources and easily interoperates with every set of scientific tools installed on our grid/cloud environment.

VI. Future developments

The most challenging part of the above architecture is ensuring our users a consistent behavior of the remote desktop protocol over as many client configurations as possible. The emerging WebRTC APIs implemented in modern web browsers [20], open up the new opportunity for a cross-platform web-based remote desktop experience, that could even avoid the need to perform the client software download to start using FARO 2.0. In the meanwhile our research is targeted towards remote desktop protocols that can enhance the features of FARO 2.0. An experimental setup has been realized with Cendio ThinLinc [21], that is based on SSH and VNC (see Figs.12-13-14), and we are planning also tests with other clientless remote desktop gateway, like for example Guacamole [22] or with remote viewer open source software like Spice [23].

As it was stressed before, the main interface is independent from remote desktop protocol, so it can be reproduced over different connection types between desktop clients and the ENEAGRID computing environment.

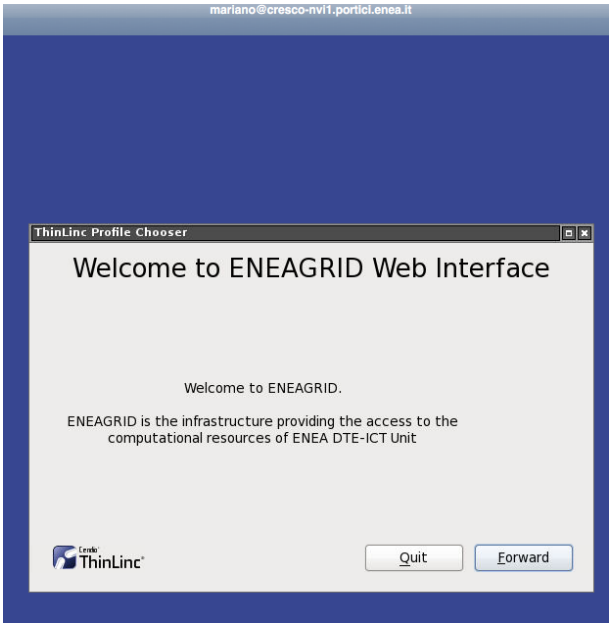

Fig. 13. Experimental FARO 2.0 setup on a Cendio ThinLinc protocol: welcome screen after logon through a browser

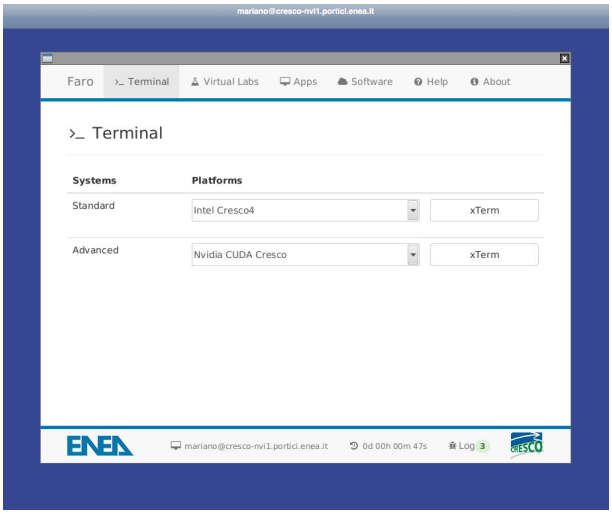

Fig. 14. Experimental FARO 2.0 setup on a Cendio ThinLinc protocol: main FARO 2.0 interface

VII. Conclusion

FARO 2.0 is a gateway for application and desktop virtualization, as well as a boilerplate to easily develop new virtual research environments. Being distributed toward users as a remote application itself, FARO 2.0 is executed on ENEAGRID and is served through an entirely managed ecosystem, thus ensuring better security for data and simpler deployment of complex scientific softwares and their updates, since everything runs on machines that are monitored and serviced by our administrators. Most part of FARO 2.0 has been developed through web technologies (HTML5, CSS3, Javascript), and thus is easy to maintain and customize, even without the need for a compile process. FARO 2.0 uses an open source implementation of NX as its remote desktop procotol, and an NX client is the only software component that users have to install to start using it. The future usage of WebRTC based clients may lead to an even leaner remote desktop experience. We believe that application and desktop virtualization constitute an excellent tool and service distribution model, that lets scientists focus more on research since relying on a high performance environment that is completely managed by our administrators.

Acknowledgment

The authors would like to thank Alessio Rocchi who was the former developer of FARO application and all the people involved in the management and operation of ENEAGRID/CRESCO infrastructure [24]. We also thank Matteo Caldera who is in charge of the computational core of the ADP Virtual Lab. FARO 2.0 has been developed in the contest of the Italian PON "Smart Cities and Communities" R&C 2007-2013 with the project "EDOC@Work 3.0 - Education and work in the cloud". Part of this activity has been supported by the ENEA-Forschungszentrum JULICH GmbH contract "Fornitura da parte ENEA di attività di modellistica molecolare".

References

[1] G. Ponti et al, *The role of medium size facilities in the HPC ecosystem: the case of the new CRESCO4 cluster integrated in the ENEAGRID infrastructure* in Proceedings of the International Conference on High Performance Computing and Simulation, HPCS 2014, Bologna, Italy, 21-25 July, 2014
[2] https://www.edocwork.it/
[3] http://www.sys-man.it/
[4] A. Rocchi, S. Pierattini, G. Bracco, S. Migliori, F. Beone, A. Santoro, C. Sciò, S. Podda, *FARO - The Web portal to access ENEAGRID Computational Infrastructure* in Proceedings of the International Workshop on Science Gateways (IWSG2010), Catania, Italy, R. Barbera, G. Andronico and G. La Rocca (Eds.), Consorzio COMETA (2010). ISBN 978-88-95892-03-0
[5] https://www.nomachine.com/
[6] https://sourceforge.net/projects/freenx.berlios/
[7] http://opennx.net/
[8] http://web.mit.edu/kerberos/
[9] https://www.openafs.org/
[10] http://www-03.ibm.com/systems/platformcomputing/products/lsf/
[11] https://azure.microsoft.com/en-us/services/remoteapp/
[12] https://www.citrix.com/products/workspace-cloud/overview.html
[13] G. Bracco et al, *CRESCO HPC System integrated into ENEA-GRID environment* in Proceedings of the Final Workshop of the Grid Projects of the Italian National Operational Programme 2000-2006 Call 1575, Catania, Italy,10-12 February, 2009 Consorzio COMETA (2009). ISBN-978-88-95892-02-3
[14] F. Ambrosino, A. Colavincenzo, G. Guarnieri, A. Funel, G. Ponti, A. Mariano, F. Palombi, G. Bracco, S. Migliori, *NEPTUNIUS: the ENEA HPC portal for multiphysics simulations*, ISC 2015, Frankfurt 12-16 July 2015
[15] http://www.afs.enea.it/project/cmast/
[16] https://code.google.com/archive/p/neatx/
[17] https://www.ark3d.enea.it/home.php
[18] http://www.afs.enea.it/project/adp/
[19] http://www.afs.enea.it/project/webcrawl/
[20] https://webrtc.org/
[21] https://www.cendio.com/thinlinc/what-is-thinlinc
[22] http://guac-dev.org/
[23] http://www.spice-space.org/
[24] http://www.eneagrid.enea.it/people/2015EneaGridPeople.html