DRAC (Distributed Remote ACess system) An On-line Open Source Project-based Learning Tool

Alexander Zlotnik, Juan M. Montero

DRAC (Distributed Remote ACess system) An On-line Open Source Project-based Learning Tool

Alexander Zlotnik[2,], *Juan M. Montero*[1],

Universidad Politécnica de Madrid, Dpto. Ingeniería Electrónica

Key words: *project-based learning, Remote and virtual laboratories, mashUp technologies, Cost-effectiveness, Pilot projects*

Abstract:

Project Based Learning is an essential component in modern engineering education. However, it is difficult to implement in a virtual simulation-based approach, and the need of expensive or specialized equipment can make difficult to use this PBL approach in remote learning. In this paper, a web-based portal is presented; it allows remote access to the resources of several PBL laboratories on Electronics, with a significant emphasis on real-time digital and analog processing and multidisciplinary interactive applications. While other approaches to remote laboratories are not general or they are based on proprietary solutions, the design of the new environment has been focused on the use of free open-source tools. DRAC is being tested on a microprocessor-based laboratory course in the first semester of 2007, allowing students to use an integrated development environment from their homes.

1 Introduction

Project Based Learning is an essential component in modern engineering education [1] [2]. However, it is difficult to implement in a virtual simulation-based approach, and the need of expensive and/or specialized equipment can make it difficult to use this PBL approach in remote on-line learning. Some of the advantages of remote laboratories [3] [4] are:

- An efficient use of the available resources (up to 24 hours a day, 365 days a year).
- It is a new flexible way of integrating practice in theoretical courses.
- It allows integration of students involved in international exchange programs.

The main purpose of the system is to provide access to software or hardware resources [5] for several students at the same time. The whole design of the system was dictated by the specific needs of several graduate and undergraduate subjects related to microcontroller programming and digital electronic design. A cost-effective mashup approach was followed: several open source technologies, not designed for interoperability, are combined in a single system using the best of their individual features. The main implementation problem was the classical one in this kind of system: glue logic.

DRAC mechanism can be summed up in the following way: one or several users connect to a server (DRAC main server or DRAC MS), which provides them a temporary and restricted connection to sub-servers (DRAC SS) or other network-accessible resources (such as RFID readers with a web interface). The only requirements for the end-users are a modern Java-enabled web browser, a VPN client and a 128Kbps (or better) Internet connection.

2 DRAC from a user's perspective

2.1 Features

- Users can be organized in teams: Several members of the same team can share files and they access the same client computer at the same time, enabling efficient collaboration.
- Usage scheduler: An automated scheduler supervises how much time each team has been accessing a specific resource and disconnects every team member if a specific usage limit has been reached. This supervision strategy guarantees fair access to limited resources.
- Connection manager: Connection data is stored, if a user closes his browser for some reason, and logs in afterwards, DRAC reconnects him to the host and resources he was using.
- A simple web-based file manager: The provided file manager enables the users to create/edit/upload/delete files and create/delete directories.
- All user files are stored on the server: therefore they are accessible both through the web interface and on the remote session on the client computer. Hence, if files are uploaded through the web interface, they become accessible on the client computer. In addition to this, user's work is always saved, a critical feature since a user might get disconnected by the scheduler if the user's session times out.

2.2 User interface

The user interface is designed with intiutiviness and easy-of-use in mind. After promting for a password, the following interface is shown to the user:

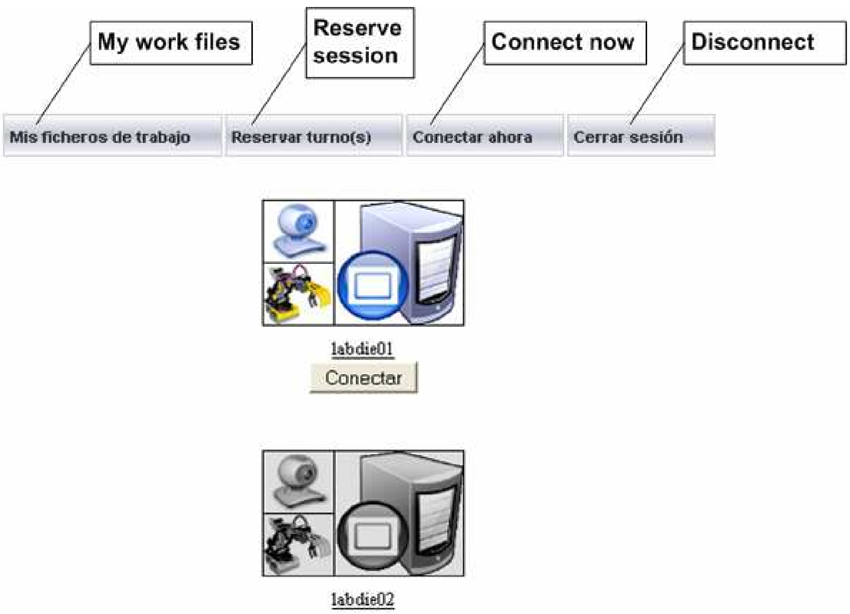

Figure 1

The "My work files" button shows the abovementioned web-based file manager. The "Reserve session" button allows reserving a resource in advance.

The "Connect now" button enables a user to connect to a resource which is not occupied. It might occur that a DRAC SS is marked as free but it is actually turned off (it might be reasonable to power off DRAC SS periodically, if the frequency of use is not high). If this happens, the DRAC MS will try to turn it on remotely when a connection to it is requested.

In Figure 1, there are several DRAC SS, although only the first one is available. Finally, the user is connected to the DRAC SS itself through a web interface:

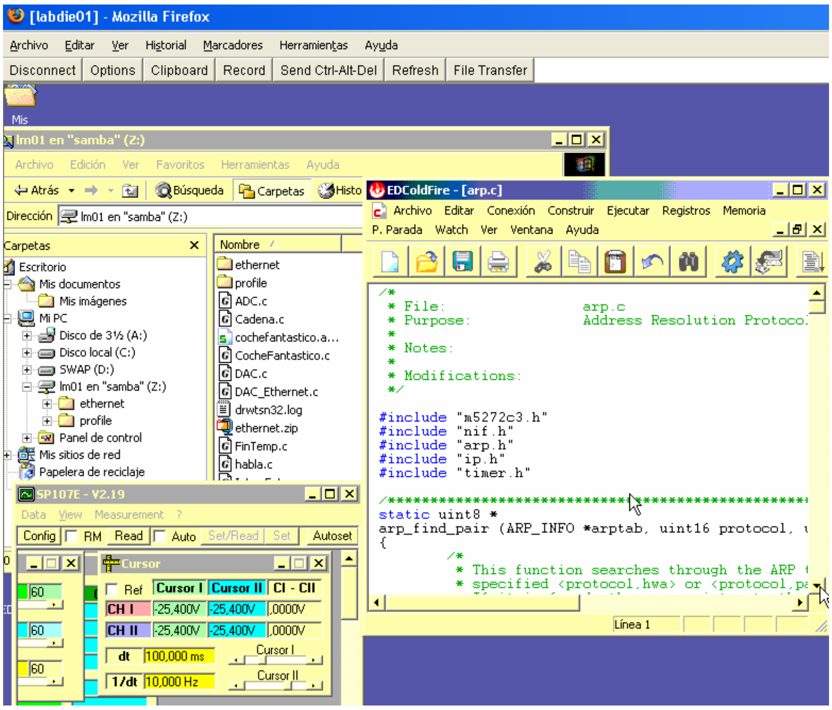

Figure 2

DRAC SS operating system can be Windows or Linux/UNIX with KDE/Gnome GUI managers. They are normally connected to highly contended resources. In this case the DRAC SS is a Windows 2000 workstation with an oscilloscope and a custom-built Motorola ColdFire microcontroller. The software packages which control them are shown on Figure 2.

It is important to note that the files accessible through the web interface (with the "My work files" button) are also directly available in the DRAC SS itself through a "network drive".

A simultaneous connection with a nearby web-enabled camera is also possible:

Figure 3

3 System architecture

3.1 Network architecture

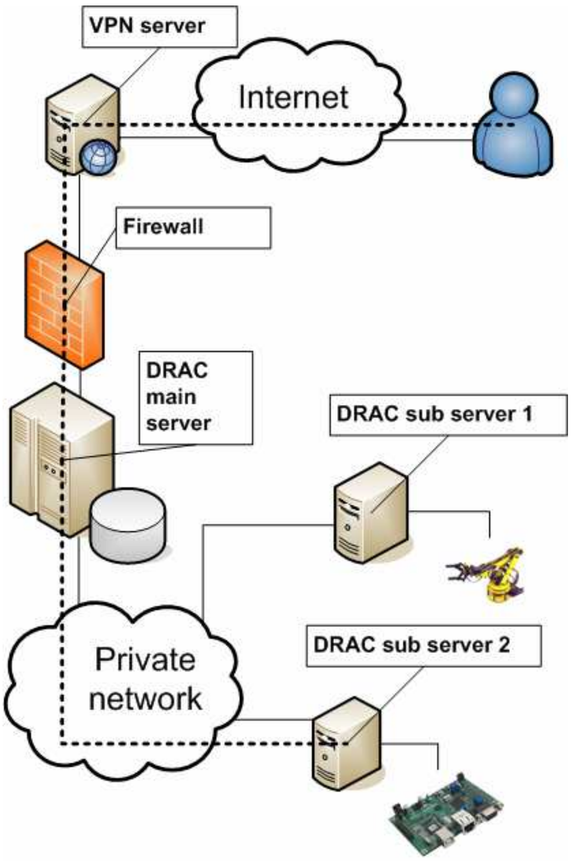

Figure 4

DRAC follows a simple n-tier approach: there is one DRAC MS and several DRAC SSs.

Before connecting to the DRAC MS, the user must use his individual private key on a VPN server (main interface to the department's network from the outside). OpenVPN is an open source VPN client which can be used for this purpose. This structure is imposed by security purposes: the VPN server is constantly monitored and regularly updated; it also stores a log of every user connection. Therefore, security requirements for the DRAC MS can be relaxed.

The DRAC MS displays a web interface, where the user must provide the username and a password associated to his team (note that the username and password provided in this second authentication stage are associated to a group; however, private VPN keys are individual). Afterwards the DRAC MS builds a NAT connection to the DRAC SS requested by the user. It is also possible to create a simultaneous connection to a DRAC SS (which can be controlled through a web interface) and a web-enabled resource connected to the private network, such as a camera with a web interface. The connection to all the aforementioned is resources is managed using iptables software package.

Should more users from the same workgroup connect to the DRAC MS, it will create NAT routes to the same resource(s) that the first user of the group requested since concurrent connections to this system are possible.

3.2 Software architecture and hardware requirements

3.2.1 DRAC SS

DRAC SS can be any kind of operating system with VNC support. We chose VNC for its multiplatform and web interface capabilities. There are several VNC servers for many operating systems with slightly different features, but most of them have the possibility to access this server using a Java applet in a web browser.

In our particular case, a Windows client was a requirement, since the EDColdFire software package – essential for our needs – was designed for this operating system. Therefore, we chose the best VNC version for this OS – UltraVNC.

Since Windows was a must-be, we also decided to make use of the Microsoft Windows Server Domain system. We were particularly interested in a very important feature – make user group's directories available on the DRAC SSs through a network-based drive in Windows Explorer. This is relatively easy to achieve if the DRAC SS is associated to a Windows Domain Server, which can be emulated with a Linux server with the Samba package installed.

3.2.2 DRAC MS

Surprisingly, hardware requirements for the DRAC MS are very modest – our prototype is running on a PII 333 with 128 MB of RAM, and it could easily support up to 10 concurrent connections.

DRAC MS' operating system is Linux Fedora Core 4. The relevant software packages installed on it are:

- Apache web server with PHP 5 scripting support module.
- MySQL database.
- iptables packet filtering ruleset program.
- Samba SMB/CIFS networking protocol module.

The web interface was built entirely in PHP using the MySQL backend to authenticate users and manage their session information. Iptables NAT route creation is handled entirely using PHP scripts. A cron daemon periodically executes a PHP script to check for overtime in user sessions, and closes them (deletes the NAT route and resets the target DRAC SS).

The DRAC MS is also a Samba Domain server, which enables Windows-based DRAC SSs to use their files on the target DRAC SS. In is important to point out that Samba is a reverse engineered implementation of the SMB/CIFS protocol, therefore it lacks some features available with real Windows Domain servers, in our case, however, their implementation was sufficient. If this feature was required for UNIX / Linux DRAC SSs, it could be easily added activating and configuring the NFS service on the DRAC MS.

Finally, the DRAC MS can send Wake-on-LAN packets to DRAC SSs in order to turn them on. This, obviously, requires a WOL-compatible card on the target DRAC SS.

4 Conclusions and future work

A web-based portal has been implemented. The system allows remote access to the hardware and software resources of several PBL laboratories on electronics, with a great emphasis on multidisciplinary interactive applications. It has been tested on a microprocessor-based laboratory course in the first semester of 2007, allowing the students to use the microprocessor platform and the integrated development environment from their homes.

Several improvements are possible, mainly those related to real numerical scalability and a better fault tolerance. In both, cases a reasonable solution seems to be that of using redundant DRAC MSs and SSs. This strategy would also introduce several non-trivial challenges and it will likely require severe changes in the software architecture.

5 Acknowledgment

We want to thank all the members of the GRIDS group and Dpto. de Ingeniería Electrónica at ETSIT-UPM for their continuous effort on higher education innovation; especial thanks to Ruben San-Segundo and Javier Macías-Guarasa. This work has been partially financed by a UPM innovation project: “Desarrollo de una plataforma para la docencia semipresencial en asignaturas masivas de carácter práctico basadas en proyectos (Project Based Learning)”.

References:

[1] Solomon, G. “Project-Based Learning: a Primer”, Technology and Learning., Vol 23 (6), 2003, pp. 20-27.

[2] Montero, J.M., San-Segundo, R., Macías-Guarasa, J., de Córdoba, R, Ferreiros, J. ,“A Project-Based Learning Approach to a Curricular Design on Electronics”, IEEE Transactions on Education, Volº 49(3), 2006, pp. 389-397.

[3] Auer, M.E. “Online Laboratory Standard Tools for Mobile Learning”, IMCL Proceedings 2006.

[4] Anido, L., Llamas, M., Fernández, M. J,.” Internet-based Learning by Doing”, IEEE Transactions on Education, Vol 44(2), (2001).

[5] Nieto-Taladriz, O., Araujo, A., Fraga, D., Montero, J.M., Izpura, J.I., “Antares: A synergy between University Education and Research, Development and Technology Innovation Groups”, Proceedings of the Colloquium on Higher Education of Electronics, 2004, pp. 23-26.

[6] San-Segundo, R., Montero, J.M., Macías-Guarasa, J., de Córdoba, R., Ferreiros, J., “Automatic Tools for Diagnosis and Feedback in a Project Based Learning Course”, Proceedings of Frontiers in Education Conference, 2005.

Author(s):

Montero Martínez, Juan Manuel, MEE (ETSIT UPM), PhD EE (ETSIT UPM)
Universidad Politécnica de Madrid, Dpto. Ingeniería Electrónica
Madrid, 28040
juancho@die.upm.es

Zlotnik, Alexander
Universidad Politécnica de Madrid, Dpto. Ingeniería Electrónica
Madrid, 28040