Use Case: OpenSource Virtualization for Educational Computer Laboratories

Lucas Barboza and Cristian Damián Pacífico

Universidad Nacional de Entre Rios, Facultad de Ciencias de la Administracion,
Msñor. Tavella 1424, 3202 Concordia, Entre Ríos, Argentina
`{lucas_barboza@outlook.com,cripac@fcad.uner.edu.ar}`

Abstract. The usual case with computing laboratories in schools and other public educational institutions is that computers are not adequately set up and maintained in order to provide an homogeneous user experience. They are configured and managed individually and it is almost impossible for managers to keep up with the latest software updates in every single PC. In other cases, some institutions cannot afford having any computers at all. By implementing a central powerful virtual server and a group of lightweight hardware computers in a server-client configuration, it is possible to offer the same user experience one gets with a desktop computer. Consequently, is it possible to reduce the equipment's maintenance effort, simplify software updates and have all computers equally configured. In this work, we define the requirements for an actual educational computer laboratory and present a possible solution to overtake this limitations through the use of a virtualization platform, which provides virtual desktops environments, along with any kind of computational device that can be used as a *thin client*.

Keywords: Computer Laboratories, Virtualization, Thin Clients, SBC, LTSP

1   Introduction

One of the advantages of using virtualization technologies is that they reduce the required infrastructure to run different environments and applications by using the same underlying hardware more efficiently. This makes it possible for organizations to implement consolidation and deploy one or more isolated environments into a single physical machine that hosts a virtual server per scenario.

Implementing virtualization also helps reduce the effort required to maintain and deploy virtual machines, since this is done centrally from the server, which frees the manager from individually configuring each physical device.

This way, a central and more powerful machine can be configured to handle the workload of a group of individually connected workstations that rely on the server for their processing work to be handled. In this schema, all that those computers do is presenting the user with a graphical interaction interface and displaying the output that comes from the server as a result of a performed

task. As consequence, user workstations needn't be resourceful anymore since their processing work is transfered to and handled by the server, what turns them into a sort of dumb terminals.

So the use of *thin* and *fat clients* comes really in handy in a server-client architecture since their main job is to transfer the user input to the server, which performs the requested tasks and returns the graphical results back to the user, all through the LAN. Like this, it is possible to help reduce the cost of implementing a computer laboratory since the price of a *thin client* device is considerably lower than that of an average PC. Also, old machines that cannot handle newer Operating Systems or whose hardware is quite outdated, can be recycled to be used as thin client terminals and become usable again.

We believe this is a feasible solution for those educational institutions which cannot afford having computer laboratories due to its cost and for those who have it but find it difficult to keep it updated and configured according to their users' needs.

In this work, we first characterize computer laboratories at the Facultad de Ciencias de la Administración of the Universidad Nacional de Entre Ríos (hereafter FCAD-UNER), in terms of didactic requirements. Then, we focus on the possible virtualization techniques and the available technologies that serve as potential solutions for the issues encountered at the computer laboratories. We did not describe the general concepts related to virtualization in depth, instead we included some clarifications of the most relevant terms. (For more detailed information, we suggest reading [6, 7, 13, 18, 19]). Next, we present a practical use case where the found solution was implemented at the FCAD-UNER laboratories. Finally, we conclude this work by presenting advantages of deploying such a solution in an educational environment and some future lines of work.

2   Computer Laboratories at FCAD-UNER

Since Computer Laboratories' PCs present their own uniqueness, their analysis and better comprehension allows for a better understanding prior to decide whether to implement virtualization solutions or not.

Next, we present a brief description of the computer laboratories available at the FCAD-UNER, their given use and the major issues pointed out by main users and administrators. This is based both on physical inspections performed on the laboratories and some ad-hoc interviews made to teachers, students and administrators from that same institution.

1. For most classes that make use of computer laboratories, prepared activities rely on the computers all being configured in the same way and having identical or at least similar software running on them. This is quite often an issue because computers are not the same in terms of purchasing methods, manufacturer, hardware and/or installed software, and their configurations differ between laboratories. In consequence, the user experience students get is not equal and depends on the computer they get.

2. Some user groups make a heavy use of the computational power of the PCs, mostly related to Computer Science or Data Analysis practices, while other groups use them for office related work and web surfing. Hence, there are situations in which computers are more stressed and require better performance while in others their processing capabilities aren't used at its maximum capacity. This leads to having different laboratories where computers' hardware and software is set according to their expected use. The result of this decision is that the total computational power is distributed into many individual machines, making it impossible to make the most out of it, as it would be if it were consolidated.

3. One of the most frequent issues for administrators are configuration changes made by students. Despite imposing restrictions, passwords and privileged permissions to standard user accounts, these control measures do not appropriately prevent some situations from arising. For example, there's the usual case where, purposely or not, users infect computers via USB sticks that contain different kinds of malware.

In short, it is important for both teachers and students to have identical or equal environment where to develop their computer laboratory practices. Also, given that not all users utilize computers for the same purpose, their software and hardware varies and most of the time these resources are not efficiently employed. Finally, keeping computers up-to-date and free of unwanted configurations demands administrators constant attention, specially where individual physical configurations are required.

3 Desktop Virtualization

The main purpose of **Desktop Virtualization** is to migrate the workstation's local processing workload to a central server, turning the workstation into a graphical user interface from where to interact with and benefit from the server's computing power. By doing so, the workstation is exempted from managing both hardware and software resources -excluding some minimum required components-, because it relies on the server for all its computational power. This virtualization scenario is possible through **VDI** (Virtual Desktop Infrastructure) or **SBC** (Server Based Computing). While VDI provides a dedicated Virtual Machine (VM) for each individual user [1], SBC consists of a central virtual machine that hosts an Operating System that accepts multiple user connections at the same time [21], giving each user a separate virtual desktop environment. Figure 1 shows the differences between both virtualization options.

Although VDI permits many customized VMs according to every user needs, in terms of management it is still necessary to supervise and configure one environment per user. With SBC, on the other hand, the server hosts one or more VMs, which in turn host a virtualized Operating System that delivers virtual desktops to a range of connected clients. Therefore, it is possible to economize the number of VMs to deploy and manage when using SBC in contrast to VDI,

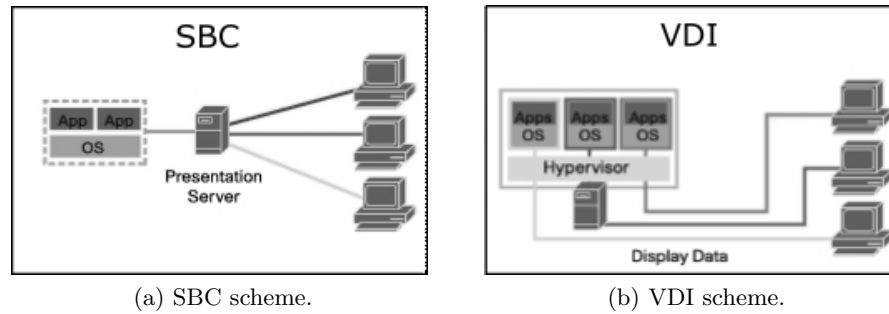(a) SBC scheme.   (b) VDI scheme.

Fig. 1: SBC and VDI different schemes.

since we only need a certain group of running VMs to provide each user with a desktop, instead of requiring dedicated VMs as is the case with VDI.

When considering the FCAD-UNER computer laboratories, we believe SBC is preferable to VDI as a possible virtualization strategy. SBC works with a single terminal server that delivers a desktop to each connected user (a private user space with its own `/home` directory); instead of having to deploy a dedicated VM per user, as it is the case with VDI. Consequently, there are fewer environments when working with SBC than with VDI, making deployment and management tasks simpler to conduct.

3.1   Thin and Fat Clients

As a result of migrating the computational power from PCs to the server, the user workstation's hardware can be drastically reduced. Or, if it is already resource-limited, it is possible to re-purpose it.

Thin clients [2, 22] are computers with minimum hardware resources that are just good enough to perform basic tasks, such as establishing a connection with a remote server; getting input from the user and sending it to the server for it to process it (getting then the results); storing limited amount of data; or displaying graphics to an output.

Fat clients [8] are similar to thin clients in that they also count on the server to perform heavier processing workload, but they are better equipped in terms of hardware, thus enabling them to handle some tasks locally; turning to the server only for the most demanding processes. While it is rare for a thin client to have more than a 1 GB of RAM or any storing devices at all, this is not the case with fat clients, which hold more hardware resources in comparison.

These devices are well-suited as client terminals in a server-client virtual desktop architecture [8], whether using VDI or SBC, since the client workstation no longer requires to have as many resources as when locally performing the processing workload.

3.2   Terminal Server Solution

As stated before, we chose SBC as a potential virtualization solution reference for the FCAD-UNER computer laboratories. Next, for the terminal server, we determined that it should meet the following requirements:

- That it is possible for the client machines to boot the terminal server session from the network.
- That the server handles the client machine's workload.
- That is not necessary to have any operating system locally installed on the client machine for it to work.
- That the terminal server solution is Open Source in order to avoid license fees.
- That the terminal server solution is compatible with Open Source virtualization platforms, such as XenServer.

After some exploratory research was done to the available technologies, we found out that the Linux Terminal Server Project (LTSP) met the requisites and so we decided to use it as the terminal server solution. LTSP is a collection of open source software that turns a normal GNU/Linux installation into a terminal server, thus enabling the use of thin and fat clients as user terminals. LTSP requires that the client device has a network card that is able to perform network booting (netboot) and at least 300 MB of RAM, for it to properly work [5]. Every connected user gets its own private space, known as chroot jail, that resides in the `/opt/ltsp` directory of the LTSP server. In order to provide a virtual desktop suitable for all devices, it is possible to configure a dedicated directory for the i386, amd64 or armhf architectures -the latter being necessary when working with Raspberry Pis as client machines [17].

4   Implementation Case

To conduct the implementation case, we used a R&D virtualization platform installed at the FCAD-UNER. The virtualization platform is made up of 2 HP ProLiant DL360 Gen 9 8SFF servers, 128 GB of RAM and 2 SSD of 300 GB capacity each; 1 HP MSA 2024" storage array, totalizing 7200 GB of storage capacity. Although this virtualization platform characteristics may seem too bulky for an average educational institution to replicate in its own computer laboratory, this is not the case since LTSP requirements are indeed very limited [5], making it accessible and affordable. We decided to use the aforementioned platform to conduct the implementation because of the ease of access to it and practical reasons solely. Nevertheless, it is totally possible to reproduce this scenario using any computer that meets LTSP server minimum characteristics.

In order to implement a LTSP server, we used XenServer to set up a VM with Ubuntu 16.04 as its GNU/Linux OS distribution. As LTSP works with Debian, Fedora, Gentoo, openSUSE and Ubuntu, we decided to use the latter because of being more used to working with it. The VM was set up with 8 GB
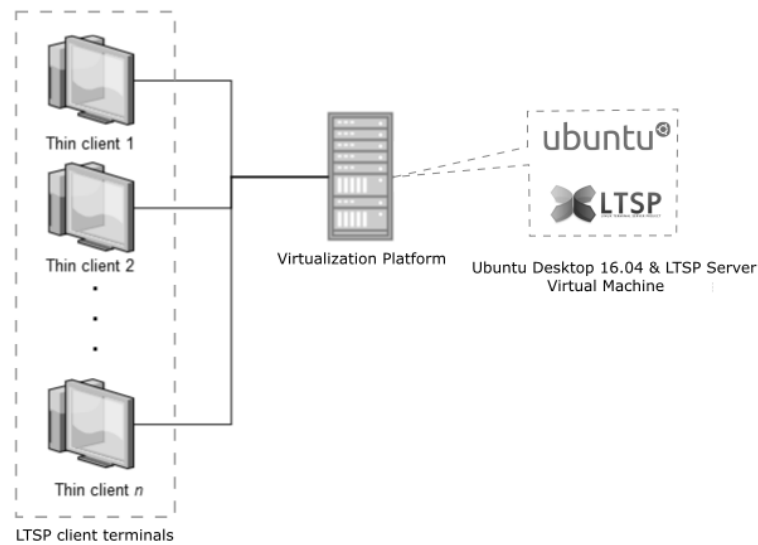
Fig. 2: Diagram of LTSP server-client implementation.

Fig. 3: LTSP working test at one of the FCAD-UNER computer laboratories.

of RAM, 6 CPU cores, 1 NIC with Internet access and a 100 GB of disk space. After installing the LTSP server and making the necessary adjustments and configurations, we connected the VM to the computer laboratory LAN. Figure 2 shows the infrastructure diagram.

We decided to conduct the test using the 16 PCs at one of the laboratories of the FCAD-UNER. The reason for that was to determine if they would all cope with the LTSP server-client architecture and could be used as thin/fat clients in a virtualized computer laboratory scenario, given that they all have different hardware characteristics and configurations. First, in order to boot the thin clients from the LTSP server, we had to enable the networking booting from each of the computer BIOS. Then, after restarting the PCs, they connected to the server and displayed the corresponding login screen, as shown in Fig. 3. We then executed some applications so as to simulate a real use case situation: Firefox Browser, OpenOffice Writer and Calc, and tried out Epoptes software

too (a computer lab management and monitoring tool[1]). We didn't notice any malfunctioning while using it so we concluded it worked correctly.

5   Benefits

We believe that educational institutions could benefit from implementing their computer laboratories with virtualization solutions, like the one we presented here, because of the following reasons:

1. **Cost Reduction**: by using free solutions such the ones proposed here, there are no license fee costs and it possible to execute any number of terminals without worrying about renewing or purchasing new licenses. Moreover, the use of thin and fat clients as user terminals means that the required investment to mount a computer laboratory is less compared to that of installing average PCs. Besides, recycled computers can be used as thin clients, obviating new acquisition costs.

2. **No Proprietary Licenses**: the GNU/Linux distributions are not only free for everyone to use but also they allow and encourage modifications, which means that changes to the software can be done with no restrictions at all.

3. **Simplified Maintenance**: by centralizing and consolidating the whole computer laboratory system into a server, or groups of servers, the management is simplified since the maintaining of the operating system and installed applications is done from a single location. What is more, it is easier to migrate, change, update, install and remove software because doing it once is enough for the whole system to adopt the new configuration. Also, since it is not necessary to have any installed operating system on the client device for it to work, by installing more than one OS in the terminal server gives the user the chance to choose what OS to launch at booting time. Besides, in case a rollback to a previous configuration is needed, using the built-in snapshot tools from XenServer (or any other virtualization server) comes really in handy. Finally, as LTSP clients don't get root access to the terminal server, all user actions are restricted to their given `/home` directory, preventing them from modifying other users desktops and the main system.

4. **Customization**: professors and teachers could freely choose their own customized desktop environment according to their lessons and needs. This way, students of a given class would all have access to the same version of applications, a certain operating system, a special set of programs and/or particular documentation; all this considering that didactic activities present their own requirements.

5. **Machine Independence**: since the workstation no longer stores any data, a user can use any available terminal (even one of his own) to connect to the server and perform their desired tasks. This facilitates the replacement of defective machines and favors the BYOD (bring your own device) movement [15].

---

[1] http://www.epoptes.org/

6. **Improved Security**: it is possible to implement SSH tunnels between the LTSP server and the connected terminals, and also restricting the communication to a specific LAN. What is more, the use of private chroot jails prevents the propagation of frequently distributed malware, since user sessions are kept isolated from each other and even from the server operating system.

7. **Green Computing** [8]: virtualization contributes to reduce the ecological footprint as it consumes less energy per user while making a better and more efficient use of the underlying hardware. In addition, the use of thin and fat clients demands less electricity compared to PCs, which not only benefits the environment but also reduces the energy expenses. Finally, thin and fat clients have a longer lifespan than traditional PCs, and considering that older computers can be recycled an turned into thin clients, this solution helps reduce E-waste too.

6   Conclusion

After successfully implementing a virtual desktop environment at the FCAD-UNER computer laboratories, based on SBC server-client architecture, using Ubuntu, LTSP and thin/fat clients; we believe it represents a convenient option for those educational institutions that find it difficult to deal with their computers' systems heterogeneity, management and updating; and also for those organizations that cannot afford installing a computer laboratory because of high costs of purchasing new PCs, but could benefit from using thin clients instead.

7   Future lines of work

Perform a comparative between different available OpenSource solutions that provide virtual desktops trough a terminal server (different to LTSP). Analyze the possibility to implement the solution using other devices apart from thin clients (smartphones and tablets). Formally perform and record stress tests in order to quantify the system QoS. Document the required training to implement and maintain such a system, as well as the necessary manuals and procedures to operate it.

References

1. VDI: A New Desktop Strategy, Dec 2006, Daniel Bev eridge, `https://www.vmware.com/pdf/vdi_strategy.pdf`
2. Thin Clients 2011 - Ecological and economical aspects of virtual desktops, Christian Knermann et al., `http://cid448u1296.hd52.hosting.punkt.de/fileadmin/user/upload/documents/PDF_files/White_Paper_EN/thinclients2011-en.pdf`
3. Install LTSP to run a thin client setup, Feb 2016, `http://www.havetheknowhow.com/Configure-the-server/Install-LTSP.html`
4. Concepts - LTSPedia, Sep 2017, `http://wiki.ltsp.org/wiki/Concepts`

5. LTSP Installation & requirements, `http://wiki.ltsp.org/wiki/Installation`
6. Operating-system-level virtualization - Davoud Teimouri, 2018, `https://www.teimouri.net/operating-system-level-virtualization`
7. Virtualización de aplicaciones, 2018, `https://inlab.fib.upc.edu/es/virtualizacion-de-aplicaciones`
8. Shalabh Agarwal, Rana Biswas, and Asoke Nath. Virtual desktop infrastructure in higher education institution: Energy efficiency as an application of green computing. In *Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on*, pages 601–605. IEEE, 2014.
9. Yassine Benkirane. *Clients légers sous Linux: LTSP (French Edition)*. eBook, 2014.
10. Ka Ching Chan and Mary Martin. An integrated virtual and physical network infrastructure for a networking laboratory. In *Computer Science & Education (ICCSE), 2012 7th International Conference on*, pages 1433–1436. IEEE, 2012.
11. Pawel Chrobak. Implementation of virtual desktop infrastructure in academic laboratories. In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, pages 1139–1146. IEEE, 2014.
12. Chang-tseh Hsieh. Strategies for successfully implementing a virtualization project: A case with vmware. *Communications of the IIMA*, 8(3):1, 2008.
13. Michael Jorgensen, Michael Fairbanks, Jason Allen Sabin, Nathaniel Brent Kranendonk, and Kal A Larsen. System and method for providing virtual desktop extensions on a client desktop, June 18 2013. US Patent 8,468,455.
14. Cao Le Thanh Man and Makoto Kayashima. Virtual machine placement algorithm for virtualized desktop infrastructure. In *Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on*, pages 333–337. IEEE, 2011.
15. Afreen, Rahat Bring your own device (BYOD) in higher education: opportunities and challenges In *International Journal of Emerging Trends & Technology in Computer Science*, volume 3, pages 233–236. 2014.
16. Harry Mangalam. Setting up an LTSP Thin client system, Nov 2010.
17. UbuntuLTSP/RaspberryPi `https://help.ubuntu.com/community/UbuntuLTSP/RaspberryPi`
18. Daniel A Menascé. Virtualization: Concepts, applications, and performance modeling. In *Int. CMG Conference*, pages 407–414, 2005.
19. Tomislav Petrović and Krešimir Fertalj. Demystifying desktop virtualization. In *Proceedings of the 9th WSEAS international conference on Applied computer science*, pages 241–246. World Scientific and Engineering Academy and Society (WSEAS), 2009.
20. Stephen Soltesz, Herbert Pötzl, Marc E Fiuczynski, Andy Bavier, and Larry Peterson. Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. In *ACM SIGOPS Operating Systems Review*, volume 41, pages 275–287. ACM, 2007.
21. Volchkov, Andrej Server-based computing opportunities In *IT professional*, volume 4, number 2, pages 18–23. IEEE, 2002.
22. Tolia, Niraj and Andersen, David G and Satyanarayanan, Mahadev, Quantifying interactive user experience on thin clients In *Computer*, number 3, pages 46–52. IEEE, 2006.