ABSTRACT

This paper investigates how Cloud computing can meet the demands of people with disabilities who occasionally use a shared computer. In this situation, customized assistive software can not be available to the user since security policies prevent from having enough privileges to change local system preferences. In order to address such issue, we discuss an open source software architecture combining a web-based remote desktop management solution with virtualization technology. This system allows disabled users to access a virtual desktop running personal assistive software solutions. Hence, the disabled user can interact with the same virtual environment from any networked physical computer via a standard web browser. In the end, we discuss the major technological issue for the achievement of such a scenario.

Categories and Subject Descriptors
K.4.2 [Social Issues]: Assistive technologies for persons with disabilities; C.2.4 [Distributed Systems]: Distributed applications

Keywords
Cloud computing; Assistive Technology; Virtual Machine; Remote Desktop; HTML5; VNC

1. INTRODUCTION

People with disabilities interact with computing devices using customized Assistive Technology (AT) solutions. Such tools include software and hardware peripherals that assist users in accessing computers [13]. In this paper, we deal with AT desktop software products that are intended for the following three types of disabled computer users: i) people with hand dexterity who have trouble using a standard mouse or keyboard; ii) users who work with screen readers to access a computer; iii) people with low vision who use magnification software. Regardless of the AT tool type, these software components need to be suitably installed and customized. It may be a difficult task to accomplish: if the disabled user cannot setup the accessibility tools according to his/her needs, an AT expert has to help him/her. This support cannot be always available due to various reasons, (e.g., economical, social, geographical, etc). Hence, for lots of disabled people using a computer can become very difficult. Moreover, whenever the disabled person uses a shared computer (e.g., in an internet point or in a campus) the local security policies prevent him/her from having enough system privileges to properly configure his/her personal accessibility tools. On the other hand, if the disabled user can perform such operations, once he/she stops using it, all the system settings have to be restored to their default configuration. So the end user has to repeat the whole software setup process every time he/she needs to manage public computers.

Nowadays, Cloud computing allows to conceive new innovative services unlike in the past [10]. This work aims to evaluate how the virtualization technology and Cloud computing can address the aforementioned issues. We discuss a distributed software architecture allowing disabled users to use a personal desktop environment deployed on a remote Virtual Machine (VM) and accessed via a standard web browser. This VM is suitably customized according to real users’ needs and it runs appropriate AT tools. Thus, the disabled user only needs an Internet connection and any networked computer with a web browser to access anywhere and any time his/her virtual desktop environments without installing additional software on the physical computer which he/she occasionally manages. Moreover, the VM setup process has to be performed only once: the task can be attended by the same disabled user or by a remote AT expert, who can support lots of disabled people living
away, as in emerging countries or in rural regions.

The rest of the paper is organized as follows. Section 2 discusses related works. In Section 3, we introduce our architecture. Section 4 deals with three case studies analysing how the proposed software infrastructure supports common assistive software desktop applications (i.e., keyboard preferences, screen magnifier, text-to-speech, virtual keyboard and bookmarklet) running on a remote VM. Section 5 concludes the paper with hints for the future.

2. RELATED WORKS

Using AT software in a remote virtual desktop according to Cloud computing requirements is an emerging topic [17], [15]. Assistive augmented browsing systems are related to the web browser and their main goal is to achieve an environment in which assistive tools can be deployed together with multimedia contents. To perform these operations, an end user has to install an additional plugin on the browser in order to make on-the-fly changes. In this scenario, plugins provide some assistive features, including screen magnifier, text-to-speech, virtual keyboard and bookmarklet [14]. Such tools are less powerful than assistive software packages which can be executed on a desktop environment and their main goal is to enhance accessibility of any website [2]. In this context, WebAnywhere is a remote web-based screen reader application allowing people who are blind to access the web from any networked computer with a modern web browser and some additional plugins (e.g., Adobe Flash) to play audio. This solution consists in a web page including a self-voicing web browser: when an end user browses a web page, WebAnywhere describes via audio the contents that are loaded.

From an innovation perspective, Cloud4all [1] and GPII [3] aim to provide a novel paradigm in accessibility using the cloud technology. Another initiative that aims to provide a Cloud platform for the development of customized services is the Distributed Resilient Adaptable Cloud Oriented Platform as a Service (DRACO) PaaS [9]. These projects aim to exploit Internet and all its information in different fields including education, employment, daily living, e-commerce, health, safety, energy sustainability [12]. Using Virtual Machines in a Wide Area Network represents the new challenge if we consider the great benefits of the remote Desktop [16], [8].

3. ARCHITECTURE OVERVIEW

In this Section, we propose a software architecture enabling users to access a VM by means of a HTML5 web interface. The proposed system only includes open source cloud computing technologies. To access the virtual desktop environment running on a VM, our solution is to establish a VNC connection via a HTML5-based VNC viewer web application that does not require any additional software. In this way, a user can access his/her VM establishing a connection from his/her browser to the HTML5-based VNC Viewer Web Application. The main components of our architecture include:

- **VNC Viewer Web Application.** It is a web application deployed in web-servers, such as Apache. It integrates a HTML5-based VNC Viewer in order to enable users to interact with VMs. Also, the web application exploits Canvas tag features in order to display remote screen content and handle keyboard / mouse events.

- **Web-WebSocket Proxy Server.** It acts as an interface between the VNC server and the VNC Viewer Web Application. The proxy receives input commands from the web client software and translates them to corresponding inputs for the VNC server. In addition, the proxy is able to process the remote screen content and send information to the VNC Viewer Web Application.

- **Web Browser.** It can be any HTML5 compatible web browser (e.g., Google Chrome, Mozilla Firefox, Safari, Microsoft Internet Explorer).

- **VNC Server.** It is a piece of software running on VMs that enables the users to control their desktop environment.

- **Hypervisor.** It is the Virtual Machine Manager (VMM) component. This software allows you to manage VMs.

The VNC Viewer Web Application has been developed according to the open source “noVNC” Project [7], which offers an open source Python Web-socket proxy server and Javascript libraries to interact with a remote VNC server. The VNC Viewer Web Application has been deployed in the Apache web server, whereas as hypervisor, we adopted Oracle VM VirtualBox [6].

Figure 1 depicts the architecture deployed in a single physical machine running a Linux OS.

![Figure 1: noVNC implementation.](image-url)

According to this implementation, three steps are required to access a remote VM: 1) the end user points the browser to the web page that contains the VNC Viewer Web Application; 2) a connection with the Web-WebSocket Proxy Server is established via socket; 3) the Web-WebSocket Proxy Server establishes a connection with the VNC server running on the VM using the RFB protocol. The Web-WebSocket Proxy Server translates VNC binary data to text data and sends them to the viewer (and vice versa). This software architecture is a working solution for starting a VNC session with a VM. It allows to view and control a remote virtual desktop by means of a web page. All the available software on a VM can be considered as a service over the network, eliminating the need to install additional pieces of software (or plugins) on the physical machine running the web browser.
4. ASSISTIVE SOFTWARE ON A REMOTE VM

In the following, three case studies are discussed to check how the proposed software architecture allows the user to interact with some AT desktop tools running on a remote VM. These applications include: i) built-in keyboard assistive preferences; ii) screen reader software; iii) screen magnification software. Such assistive software solutions have been installed and customized on a Microsoft Windows XP Professional VM that has been accessed using the designed web interface. The remote desktop environment comes with some pieces of AT software. More specifically, we have worked with Microsoft Magnifier as basic screen magnification program, whereas MouseKeys and FilterKeys options have been tested as keyboard AT preferences. In this context, MouseKeys allows the user to employ the numeric keypad to control the mouse pointer, while FilterKeys options enables the user to customize the rate at which a key is repeated when he/she holds it down. On the same VM, we also worked with NVDA open source screen reader software. Our purpose was to manage these pieces of software without having to setup any software on the physical, client computer running the web browser. Our results have been disappointing, as shown below.

- The keyboard accessibility preferences do not work; the user cannot move the remote mouse pointer and he/she is not able to have the FilterKeys option without enabling the same assistive preferences on the physical machine. This forces the user to modify local system settings, that is an unacceptable constraint.

- The VNC-based remote desktop solution does not support any form of audio output. Therefore, the user cannot work with a screen reader running on the VM.

- The magnifier correctly works. As depicted in Figure 2, the HTML5 Canvas tag allows you to view the enlarged desktop environment within a browser, and the color inversion option is supported. In addition, we have also test ZoomText screen magnifier on the Windows XP VM. At present, this accessibility tool conflicts with the VNC session, so it does not support our web interface.

To allow the end user to work with some keyboard accessibility tools and provide remote audio streaming, two specific software solutions has been added in our architecture, as it will be described in following.

4.1 Keyboard Accessibility Solution

As shown in the Figure 3, by using JavaScript technology, we have developed the keyboard accessibility preferences mentioned previously. Such features aim to detect the low level keyboard and mouse events occurring into the web browser window.

4.2 Audio Redirection System

Since the VNC technology does not support audio redirection feature, we suggest a solution to redirect all the VMs sound effects over the network. This live audio streaming has to be played on the client (physical) computer’s speakers so that the end user can hear the remote audio output through the same web page displaying the VM desktop environment. According to the hypervisor setup, VMs support a virtual sound card and the audio output is managed by the PulseAudio subsystem running on the Linux physical server. The basic idea is to compress this signal in order to send it to IceCast2 audio streaming server [5]. Such operation requires GStreamer pipelines [4] and Lame MP3 encoding tools. Finally, the same streaming server handles the audio outputs and transmits them over the network. This process is summarized in Figure 4.
To remotely hear the audio streaming, an audio player has been added to the VNC viewer web application and it does not require any additional features on the browser. In relation to computer AT tools, the audio redirection system meets the demand of blind people: for instance, these users can interact with a powerful, fully customized screen reader running on a VM and navigate their usual desktop environment by using the keyboard. At present, such pieces of software work correctly but they are not usable due to excessive delay time.

5. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

The proposed AT solution is intended for disabled people who rely on customized assistive software. The basic idea is to run these desktop applications on VMs and access them via a web browser. By using any networked computer with a standard web-browser, disabled users can manage anywhere and anytime their usual assistive applications and preferences, without having to install additional software or plug-ins on physical machines they use. This enables disabled users to occasionally work on shared computers. In order to provide these on-demand services, a Cloud-based open-source software architecture has been discussed. Our prototype exploits VNC to remotely control VMs, whereas a web-based remote desktop solution enables the users to interact with their assistive software running on a VM.

Nevertheless, using assistive software on a remote VM via an HTML5 VNC web-based client is not trivial at all. We experienced several assistive software and we understood that I/O is the main issue that needs to be overcome. In this paper, we specifically discussed how to properly adjust the keyboard accessibility tools and the remote audio redirection in order to enable disabled users to remotely control their assistive software in a VM via a standard web browser.

Cloud computing in the perspective of assistive services is at early stages. Currently, major obstacles include: i) the inability to optimize most of the existing software; ii) the poor remote desktop performance. Moreover, assistive Cloud-based services have to be developed considering a distributed, autonomic, performance, secure, and scalable Cloud architecture. In future works, we plan to investigate the adoption in our scenario of an alternative solution to VNC, i.e., Microsoft Remote Desktop Protocol (RDP), and we intend to develop assistive services considering the Infrastructure as a Service (IaaS) CLEVER [11] middleware.

7. REFERENCES