CLOUD-X
Remote Desktops and Applications through Web Browsers

Edgar Fabián Hernández-Ventura and Jorge Buenabad-Chávez
Departamento de Computación, CINVESTAV-IPN, Ciudad de México, D.F., Mexico

Keywords: Remote desktops, Web browsers, Cloud computing.

Abstract: Cloud computing is gaining general acceptance and we believe it will broaden its user base significantly once users can run their favourite applications in the Cloud *through web browsers* and *with the same interface* that each such application is used in a desktop or laptop computer. This paper presents CloudX, a new architecture based on web browser techonologies for the X Window system. X Window is the *de facto* standard for window display in Unix-like operating systems. CloudX translates X Window commands to web browser display commands using AJAX and SVG (Scalar Vector Graphics) technologies. CloudX does not require any extension to the X Window system nor any plugin to the web browser.

SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

There has been a lot of hype around Cloud computing in recent years, both in industry and in the research community; and rightly so. For small and medium enterprises (SMEs) and individual entrepreneurs alike, Cloud computing is quite the panacea in IT. Having access to a sound IT infrastructure, at a reasonable price, with 24/7 or so availability, and immediate use from anywhere and any device, fosters both creativity and entrepreneurship. Certainly, the cost and hassle of deploying and maintaining a solid IT infrastructure was a hindrance for many a good idea to develop.

We think the user base of Cloud computing will broaden significantly once users can run their favourite applications in the Cloud *through web browsers* and *with the same interface* that each such application is used in a desktop or laptop computer.

It is possible to run applications with a rich GUI remotely and with good performance using remote desktop technology such as *Cytrix* (Citrix Inc., 2010b) and *NX* (NoMachine, 2010). However, doing so is not as flexible as doing it through a web browser. Using remote desktop technology would involve installing a *client* software in each computer and mobile device with which a user works, and it may not be available for some software-hardware platforms. Also, its installation may require privileges (c.f. *Ipod, Ipad*) that users may not have or may not want to exercise for lack of relevant knowledge.

This paper presents Cloud-X, a new web-browser-based architecture for the X Window system. X Window is the *de facto* standard for window display in Unix-like operating systems. It consists of a general, hardware-independent set of commands to build GUIs and a network protocol for communication between an application and remote display hardware. Cloud-X translates X Window commands to browser display commands using AJAX and SVG (Scalar Vector Graphics) technologies. It does not require any extension to the X Window system nor any plugin to the web browser.

The remainder of the paper is organized as follows. Section 2 presents background material to X Window and SVG. Section 3 presents the design of Cloud-X. Section 4 presents related work and we conclude in Section 5.

2 BACKGROUND TO X WINDOW AND SVG TECHNOLOGIES

2.1 X Window (X11)

"The X Window system [typically referred to as X11 for its current major version being 11] is a computer software system and network protocol that provides a basis for graphical user interfaces (GUI) for networked computers. It creates a hardware abstraction layer where software is written to use a generalized set of commands, allowing for device independence

Fabián Hernández-Ventura E. and Buenabad-Chávez J..
CLOUD-X - Remote Desktops and Applications through Web Browsers.
DOI: 10.5220/0003611302340237
In *Proceedings of the 6th International Conference on Software and Database Technologies* (ICSOFT-2011), pages 234-237
ISBN: 978-989-8425-76-8
Copyright © 2011 SCITEPRESS (Science and Technology Publications, Lda.)

and reuse of programs on any computer that implements X" (Wikipedia, 2011). It was developed in 1984 at MIT by Bob Scheifler and Jim Gettys (Scheifler et al., 1988).

The basic idea of the X11 protocol is as follows, see Figure 1. Applications are clients that request an X server to display their interfaces through requests sent over a network. The server executes the requests in the display hardware. The server also captures input events from the user both through the mouse and through the keyboard, and sends them back to the application over the network. The application responds to events by sending more requests to the X server, and so on.

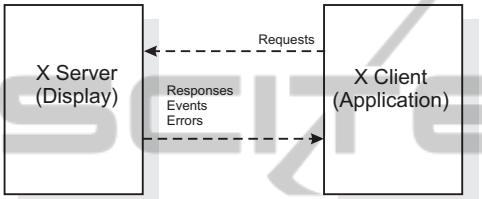

Figure 1: X11 base protocol.

In PCs and workstations using a Unix-like operating system, X11 works the same: a client application and the X server run completely separate, but on the same machine.

2.1.1 Display Capabilities

The server handles 127 different request codes, corresponding to the core X11 protocol. The client sends requests to create, manipulate and destroy server resources. These resources will affect the GUI displayed to the user and include: windows, pointer, keyboard, fonts, pixmaps, graphical contexts and palettes (colormaps). There are also requests to manage window properties, text manipulation and to subscribe to events generated from the user.

2.1.2 Event Capabilities

The event capabilities of X11 include: cursors management, resource management, color mapping, interclient communication (the ability of different X11 applications to communicate with one another) and input and output buffering, among others (Brain, 1992).

2.2 SVG

SVG (Scalable Vector Graphics) is a family of specifications in XML for describing two-dimensional vector graphics, both static and dynamic. The specification has been under development by the World Wide Web Consortium (W3C) since 1999 (SVG Working Group, 2010a).

Being defined in XML text files, SVG images can be searched, indexed, scripted, compressed, and created and edited with a text editor. There are also drawing applications that support SVG file formats like Inkscape (Inkscape, 2011).

All major modern web browsers support and render SVG markup directly with some limitations. It is expected that in 2011 all major browsers comply with at least 80% of the specification of the SVG test suite (SVG Working Group, 2010b).

2.2.1 Display Capabilities

SVG graphical objects can be vectorial graphics, raster images (PNG, JPEG) and text. Graphical Objects can be modified in several ways while being displayed. They can be grouped together, the style with which they were drawn can be changed, they can be transformed in size, rotated, etc. This allows interesting and complex ways of display and animation, making SVG an excellent tool for web design and visualization.

2.2.2 Events Capabilities

SVG images are composed of graphical objects. These objects offer different ways to interact with the user. They can have hiperlinks to new pages or generate events by change of focus, mouse click, scrolling or zooming.

Each event is handled by a defined function that may run a script, or perhaps start, stop or alter an animation. This is very important for the implementation of Cloud-X. There is no need to develop a new mechanism for mouse or keyboard input. We just use the event handlers already implemented for SVG in the browser with no additional plugins required.

3 CLOUD-X

Cloud-X was conceived out of the need for visualizing already developed desktop software with rich GUI, but remotely through the cloud. In our search, we observed that the web browser is the most widely used software to connect to the cloud, and was already installed in almost any device (including tablets and smartphones).

Although different technologies are available for drawing in web browsers, like Java applets or Flash, we wanted our software to be as non-invasive as pos-

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

sible, and also to be standard complying. After studying SVG, we made it our choice as we found it ideal for displaying interfaces, and because it is based on open standards. The latter guarantees (to some extent) long term support and usability in browsers that implement those standards. Almost any major web browser has SVG support or plans to implement it in the short term. This eliminates the need of additional software like Oracle Java, Adobe Flash, Microsoft Silverlight, etc. All we got left to do was the communication part.

Investigating the way Linux displays its graphics, we found out that X11 already catered for remote display. Both ends of the connection were already there: the applications already programmed to display their interfaces remotely and web browsers with features to display them. However, the communication cannot be completed without software that translate messages between X11 and SVG.

Cloud-X makes such translation. It is a web application programmed in Java and uses Tomcat Web Server to provide dynamic web pages to an application's GUI using SVG and AJAX.

3.1 Overview

Figure 2 shows Cloud-X. Recall that: X11 works with a client/server model, an application is the client, and when an application requires to display a GUI it sends a series of X11 commands to an X server. Cloud-X translates them and writes an SVG file. The file is created in a web server to make it available to the web browser which then displays its content. In the web browser, the user will interact with the GUI and generate events. Cloud-X translates these events to X11 events and sends them back to the application.

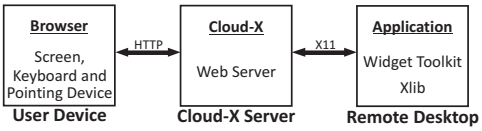

Figure 2: Cloud-X Architecture.

When only part of a GUI needs to be updated (for instance edited text or closed subwindows), instead of sending a completely new SVG file, AJAX is used to refresh immediately only that part in the user's web browser. This is very convenient as the browser only has to keep track of the currently displayed graphical objects, leaving to Cloud-X the task of managing what objects to show, and what objects to hide or eliminate from memory. Cloud-X is composed of four modules: the Access Controller, the Resources Manager, the Events Manager and the Communication Manager.

3.2 From X Window to SVG and Vice Versa

Cloud-X will map X Window commands and events into "equivalent" SVG commands and events and vice versa, as outlined shortly. Assume a user connected to a remote web server through a web browser, and about to click on an icon in order to start running the Emacs editor on a Linux remote host. The Emacs GUI interface will be displayed within the web browser. (We are ommitting some details of the communication: the initial connection to the remote web server, in which Cloud-X runs, and the communication to show the first display to the user.) The following events will take place:

1. Cloud-X opens a secure shell channel with the remote host where Emacs will run, and starts an X11-forward session so that the display output of Emacs will be forwarded to Cloud-X. (Note that both the web server, Cloud-X and Emacs may run in the same host.)
2. Emacs (through Xlib) starts the handshake protocol with the Cloud-X server, which involves asking the characteristics of the display and the X server it is dealing with, e.g.: colours available, display size, etc.
3. Cloud-X sends the web browser a simple SVG file for the web browser to start displaying a blank GUI and thus overlap some processing.
4. Emacs will then display its GUI, which involves the following. It will first send X11 requests to create the necessary resources in the X Server, actually the Cloud-X server. Emacs will then send a request to show its GUI (XmapWindow). On receiving this request, Cloud-X will generate a new SVG file with the GUI and will push it into the web browser.
5. Among other events, Emacs will request from X server to be informed of keyboard and mouse events, for instance, when Ctrl-S is entered. For each event, Cloud-X will generate a handler function that is written in an SVG file and that is pushed into the web browser. When the user interaction involves any such event, the relevant handler function will send a message to Cloud-X, which will translate it into an X11 response to send to Emacs.
6. Emacs will eventually update part of its GUI (for example, displaying the `File` menu), sending the

relevant X11 requests. Cloud-X will parse each request, translate it into a single update to the SVG file (in the web browser) corresponding to the entire Emacs GUI, and then will push each update into the web browser. The latter will then update the user display.

We are currently implementing the above functionality. After trying it out, we will design and implement the functionality relating to displaying streams (sound and video), printing and file sharing.

4 RELATED WORK

The purpose of our work is to use web browsers to display rich GUIS of applications running remotely. The work more similar to ours that we know of is the Broadway extension to the GTK+ toolkit library developed by Alexander Larsson (Larsson, 2010). Broadway extension is only for GTK+ software. Cloud-X works for any X11 application including GTK+, Qt, MOTIF, etc.

Rich GUI applications can be run remotely through Remote Desktop technology, but not within a web browser. There are various Remote Desktop clients available. NX is an optimization of the X Window system and protocol that connects to Linux desktops (NoMachine, 2010). Virtual Network Computing (VNC) has multi-platform compatibility given by the underlying protocol, the Remote FrameBuffer (RFB) protocol, which works directly with the screen buffer and just sends the raw pixel data.

The Remote Desktop Protocol (RDP) (Microsoft, 2010) and Independent Computing Architecture (ICA) (Citrix Inc., 2010a) are proprietary network protocols for the Microsoft Windows operating system. The commercial products Citrix XenAppp client uses ICA (Citrix Inc., 2010b). ICA and RDP send `GDI+` commands in ciphered packages through a TCP/IP network. `GDI+` (Graphical Device Interface) is the core Windows API for GUI drawing.

All these technologies require certain resources and system permissions to install their clients, and as mentioned earlier, they do not show the display within web browsers.

5 CONCLUSIONS AND FUTURE WORK

Web browsers run on a wide variety of devices and operating systems and will soon be compliant to the SVG standard. Through Cloud-X, web browsers will be able to display rich GUIs of X11 applications running remotely. Cloud-X offers two main advantages: i) immediate use of hundreds of applications already developed for Unix-like operating systems based on Xlib, and ii) development of new Cloud applications without the need to develop new display technology.

Cloud-X can also be used as the base technology to offer a complete operating system in the Cloud. When a new user needs an account and resources to work with, the system administrator can install a copy of Linux in a powerful server with all the software required. The user can then access this new desktop environment from any device with a web browser, eliminating the need of providing each user with a powerful machine.

ACKNOWLEDGEMENTS

Edgar Hernández-Ventura would like to thank CONACyT for the scholarship granted to carry out his MSc studies.

REFERENCES

Brain, M. (1992). *Motif programming: the essentials–and more*. Digital Pr.

Citrix Inc. (2010a). Citrix software. Retrieved from: http://www.citrix.com/site/SS/downloads/index.asp.

Citrix Inc. (2010b). Citrix xenapp. Retrieved from: http://www.citrix.com/English/ps2/products/product.asp?contentID=186.

Inkscape (2011). Inkscape. Retrieved from: http://inkscape.org/.

Larsson, A. (2010). Gtk3 vs html5. Retrieved from: http://blogs.gnome.org/alexl/2010/11/23/gtk3-vs-html5/.

Microsoft (2010). Remote desktop connection software. Retrieved from: http://www.microsoft.com/windowsxp/downloads/tools/rdclientdl.mspx.

NoMachine (2010). Introduction to nx technology. Retrieved from: http://www.nomachine.com/documents/intr-technology.php.

Scheifler, F., Gettys, J., and Newman, R. (1988). *X Window system: C Library and protocol reference*. Digital Press Newton, MA, USA.

SVG Working Group (2010a). Svg wiki. Retrieved from: http://www.w3.org/Graphics/SVG/WG/wiki/Main.

SVG Working Group (2010b). Test suite overview. Retrieved from: http://www.w3.org/Graphics/SVG/WG/wiki/Test_Suite_Overview.

Wikipedia (2011). X window system. Retrieved from: http://en.wikipedia.org/wiki/X_Window_System.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS