Remote Access through Operating System Virtualization

Pranali Jambhulkar, Urmila Shrawankar

Abstract— Operating System (OS) virtualization refers to the abstraction of an operating system from any underlying hardware architecture. OS virtualization is needed as it provides feature of transparent migration of applications. Ubiquitous computing where computing is made to appear everywhere and anywhere. The infrastructure of ubiquitous computing which exploits virtualization to make computing. In order to enable ubiquitous environment and servers to be shared the application of various operating systems with the desktop of user. Using the ubiquitous environment, the applications could be run in the host system without installation. The key concept of the system it to operate the desktop of user by any portable devices like smart phones or tablet through web browser irrespective of the location of user.

Index Terms— internet, mobile terminal, operating system virtualization, personal computer, ubiquitous, web browser.

1 Introduction

THE current scenario in computer science is considerably shifting towards Internet technologies and hence the concept of cloud computing has become more popular. The abstraction of the maintenance issues from the device has been the main motive behind this. In this upcoming age of cloud-based software, applications live on the cloud as services that can be accessed with a web browser. The services consist of data, computation and other resources that can be at anywhere in the world. Currently Software as Services is a software sharing model in which applications are hosted by a service provider and made available to clients over the Internet. The software's are stored in the warehouses, applications, which are downloaded on demand, run on a host system and the agents used by customers become an input and display device only. However the system needs to download the required software configures it and then only executes it. Moreover the system must support the operating system.

As the computing is becoming ubiquitous the diversified desktop computers, running the same OS, and does not provide the user with their personalized desktop environment, which includes personal documents, recurrently-used applications and customizations. A laptop is a way out, but the user has to carry a heavy and bulky device. So there is another equivalent changeover presently taking place is: mobile devices. These are becoming an important application platform and an opening to the Web.

Latest web applications are motivating towards fully functional desktop software such as e-mail clients and productivity apps. In this method, a web browser is typically in use as the running platform with the collaboration from a remote user desktop; and then, the user can access his/her personalized operating environment through the available web browser by any handy device. However, the huge existing desktop software cannot be used directly in this method.

The structure of this paper is as follows. In Section II provides some background on few existing solutions in the domain. In Section III, the architecture of the propose system is described in brief. Finally, Section IV concludes the paper.

2 Related Work

The Alcatraz scheme [1] [3] provides a remote execution environment on the Linux operating system. File modifications are redirected to a cache hidden to other processes. Though, it requires that applications to be installed on the host system, which is not essential for the virtual execution environment. The progressive deployment system (PDS) [2] [3] is a virtual execution environment, designed distinctively for deploying software on demand. It mainly focuses on the files sharing and delivering. Applications Virtualization [3] framework is created to execute Windows application. This model is implemented in the user-level rather than kernel-level. Three optimized virtualization strategies are implemented that includes virtual registry, file visit virtualization and system objects virtualization to organize isolate applications from OS and from other application partially.

The Featherweight Virtual Machine (FVM) [3] [4] is OS level virtual machine architecture. This architecture isolates applications from one another and from the host machine using a technique "namespace" virtualization. It is similar to Alcatraz, which requires the applications to be installed on the host system. Furthermore, it stores the registry data associated to the applications on the host system, which is not required in the virtual execution environment. Shuttle [5] which allows inter-application interactions within and across OS-level virtual machines on different versions of Windows OS. It inter-

- *Pranali Jambhulkar, Research Scholar, Department of Computer Science and Engineering, G.H. Raisoni College of Engineering.Nagpur, India. E-mail: vinyj10@gmail.com*
- *Urmila Shrawankar, Department of Computer Science and Engineering, G.H. Raisoni College of Engineering.Nagpur, India. E-mail: urmila@ieee.org*

cepts system calls related to IPC, file, registry and process for invoking inter-application interactions. On-Demand Personalized Software [6] is also based on an OS-level virtualization technology that converts desktop application to on-demand software across internet. The user's data and applications and configurations are made handy. OS-level virtualization environment is covered on top of the local machine's OS where each personalized application runs. This environment intercepts some resource-accessing APIs, as well as those accessing the system registry, files/directories, and environment variables, from these applications, and redirects them to the definite storage location rather than the local host.

The portable desktop system [3] [7] stores the user data and configurations on a portable USB device. This allows a user to run software on any compatible computer over the Internet without any setting. It redirects the files visit in the kernel-level. The virtualization-based SaaS system (vSaaS) [3] [8] is an architecture that allows users to access different software, which are deployed on it. The user can access software transparently whenever required without any limitation on the client operation system. CITRIX [6] [9] is a framework that allows a range of isolated computers to unite to a Windows NT terminal server to access a powerful desktop and applications.

MeghaOS [10] is cloud based operating system that allows user to access applications which is stored completely or partly on the Cloud. MeghaOS runs in an ordinary web browser from anywhere, any platform, any device without installation or browser plug-in components. It is designed using simple web technologies like HTML, CSS, JavaScript, XML, and PHP. Cloud Terminal Operating System CTOS [11] just runs one application which is Web browser, and the Web-based applications to convene all the daily needs of users. It abandons the traditional desktop applications, and makes some unique design to optimize Web-based applications. ServiceOS [12] a multi application OS on which web applications and conventional desktop applications unite. Services are deployed in the cloud and cached on the client. It gives controlled access to all system resources, enables offline operations, and provides suitable protection across application principals in terms of both access and usage. It also supports the same software and content sharing model as web applications, which enables OS, device, and location independence. The Illinois Browser Operating System (IBOS) [11] [13] is very similar to CTOS. IBOS is a browser Operating System is based on microkernel [13] [14], [15] L4: Ka [16]. A new layer is designed above the kernel is Browser Abstract layer that helps IBOS handle the Web-based applications. The capacity is inadequate when commerce with the Web-based applications. Chrome OS [10] [11] [13] [17] is designed and implemented by Google Inc [18]. It is Web based Operating System, same as the CTOS. It allows users to access quickly Internet and have their applications and data stored on the Internet. Web OS [19], is designed to provide web application development framework using browser as interface. It works with a URL-based file system that allows a Web-based application access files from the OS provider's server online through a domain-name-system query. Web OS also uses a location-free resource naming system that allow applications make calls to services and resources on remote severs. The comparison of various Cloud based as well as browser operating system is shown in Table I.

Table I: Comparison of the existing web based operating systems.

Parameters	CTOS	Google Chrome OS	ServiceOS	IBOS	MeghaOS	Web OS
Device Specificity	Yes	Yes	Yes	Yes	No	No
Type of Service	Cloud Based	Largely Web Based Applications	Cloud Based	Cloud Based	Cloud Based	Native Applications Essentially Web
Plugins Requirements	Requires Plugins	Preinstalled with the Chromium Browser	Plugins Required	Plugins Required	No Plugins Required	Requires Flash and other plugins
Programming Environment	JavaScript	Need Chromium API's	Need API's	Need IBOS API's	Simple JavaScript Files	Qooxdoo
Large Computing Applications	No	Not Very Suitable	Not Very Suitable	Not Very Suitable	Possible with Computational Offloading to Cloud	No
Mobile Optimization	Not Suitable	Not Suitable	Not Suitable	Not Suitable	Suitable as Mobile Version is Available	Partly Suitable

n.

The Personal Server aims to prevail over the primary inadequacy of cell-phones, PDAs, and laptops. These devices are small enough to carry and also the displays are too small to easily use. Thus, Personal Server needs to be connected to another device that has a big screen.

Remote Desktop [21], included with Windows XP, allows users to hook up to their computers across the Internet from any computer. The drawback of Remote Desktop is that users need to leave their computers turned on every time. Besides, this computer is a single point of failure and the user has no choice about the OS.

SoulPad [6] [22] is a solution that is based on moving an auto configuring operating system alongside with a perched virtual machine on a small handy device. Through this approach, the PC boots from the device and resumes the virtual machine, therefore giving the user access to his personal environment, as well as previously running computations. A SoulPad-like solution is Desktop on Keychain (DoK) [6] [23] where it uses the local machine's installed operating system i.e. nothing but Windows and runs VirtualPC there. A commercialized invention is Moka5 [6] [24] LivePC, which contains the whole thing desirable to run a virtual computer: an operating system and a set of applications. Users can access LivePC through a network server. In this case, a fundamental variation is that server can provide a VM image "diff" during streaming so that the end user will not incident unjustifiable delays on system start-up.

Some latest commercial contributions challenge to support personalization of anonymous PCs at OS-level. For instance, Migo [6] [25] allows users to hold own settings and files on a USB flash key. It only saves personalized data into the USB storage, not the applications themselves. U3 [6] [26] presents an advance arrangement of portable applications so that software should be rewritten for portability. Another analogous way out is Ceedo [6] [27], but there is no manuscript of its performance technologies. And it only provides the personal translation that stores the applications on the portable device; no any networked version is implemented, while the versions are implemented for both LAN and WAN cases.

Unlike these existing approaches, the proposed solution is based on the lightweight virtualization. Based on the virtualization technology, this paper presents a solution to enable a user to access his/her personal computer i.e. user's own desktop across the Internet via web browser on any handy device like Smartphone's or PDA's. In this approach, the user's data and applications and configurations are made moveable; every personalized application runs in an OS-level virtualization environment layered on top of the user's desktop OS. In view of that, the storage capacity required is much lesser, and the performance overhead introduced by virtualization is almost negligible.

3 Proposed Work

With the advent in high-speed Internet technologies, the concept of cloud computing has become more popular. But the problem with cloud is that the user needs to create and manage it. Another problem is that mobile devices such as smart phones and PDAs (personal digital assistants) have their own operating system. If the user wishes to access application of different operating system no solution is available. These devices are becoming more and more powerful and are increasingly used for various tasks and browsing web pages. However, there is no facility of such software that can access our own desktop through mobile or any handy device. These problems are overcome in the proposed system. Therefore, it is desirable to tailor information access on mobile devices from user desktop.

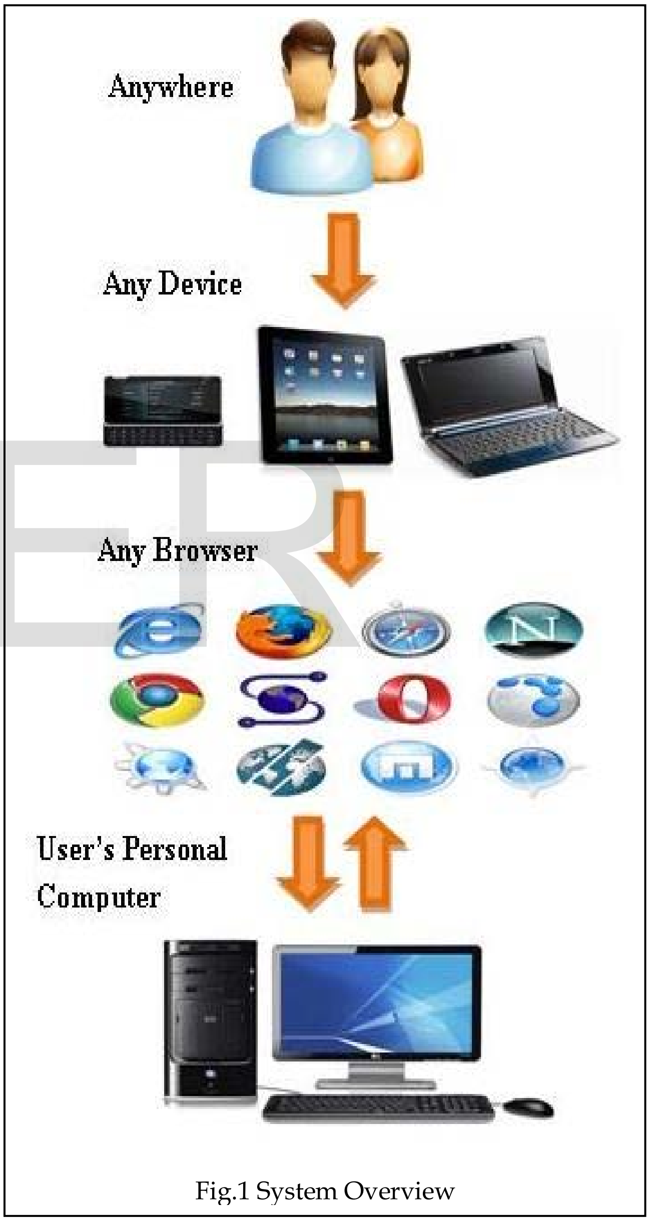

Fig.1 System Overview

4 Conclusion

There has been rapid surfacing of the World Wide Web as a platform for existent applications, which have led to increase in new opportunities for development of software web based applications. This would helps to integrate the usability of conventional desktop applications and the latent of the World Wide Web. This paper thus summarizes all the concepts associated with the development of operating system virtualization and various Cloud as well as browser based operating system. This would helps to integrate the usability of conventional desktop applications and the latent of the World Wide Web. This paper thus summarizes all the concepts associated with the development of operating system virtualization and various Cloud as well as browser based operating system.

Virtualization is rapidly becoming a common piece of system infrastructure, particularly in high-end servers. Understanding the impact of various virtualization technologies and techniques on the run times of workloads is important, particularly for large parallel cluster applications which may require deterministic synchronization in order to run efficiently. This paper proposed concept towards supporting the application interactions in an OS-level virtualization system through browser. The applications can be accessed anywhere, anytime through any handy device like Smartphone or PDS.

REFERENCES

[1] Z. Liang, V. Venkatakrishnan, and R. Sekar, "Isolated program execution: An application transparent approach for executing untrusted programs," in Proceedings of 19th Annual Computer Security Applications Conference, December 2003. DOI :10.1109/CSAC.2003.1254323.

[2] B. Alpern, J. Auerbach, V. Bala, T. Frauenhofer, T. Mummert, and M. Pigott, "Pds: A virtual execution environment for software deployment," in Proceedings of the 1st International Conference on Virtual Execution Environments, 2005. DOI:10.1145/1064979.1065004

[3] Hailei Sun, Tianyu Wo, "An Optimized Approach to Virtualize an Execution Environment for Windows Applications", International Conference on Cloud and Service Computing,2011,DOI:10.1109/CSC.2011.6138521.

[4] Yu, Yang, Fanglu Guo, Susanta Nanda, Lapchung Lam and Tzicker Chiueh, "A Featherweight Virtual Machine for Windows Applications," Proceedings of the Second ACM/USENIX Conference on Virtual Execution Environments (VEE'06), June,2006.DOI:10.1145/1134760.1134766.

[5] Zhiyong Shan, Xin Wang, Tzi-cker Chiueh, "Shuttle: Facilitating Inter-Application Interactions for OS-level Virtualization" , IEEE Transactions On Computers , 2012 ,DOI :10.1109/TC.2012.297.

[6] Youhui Zhang, Gelin Su, and Weimin Zheng, "Converting Legacy Desktop Applications into On-Demand Personalized Software" , IEEE Transactions On Services Computing, Vol. 3, No. 4, October-December 2010, DOI: 10.1109/Tsc.2010.32.

[7] Y. Zhang, X. Wang and L. Hong, "Portable Desktop Applications Based on P2P Transportation and Virtualization," Proceedings of the 22nd Large Installation System Administration Conference, November, 2008.

[8] Liang Zhong, Tianyu Wo, Jianxin Li, Bo Li, "A Virtualization-Based SaaS Enabling Architecture for Cloud Computing," Sixth International Conference on Autonomic and Autonomous Systems, pp.144-149, 2010, DOI:10.1109/ICAS.2010.28.

[9] CITRIX, http://www.citrix.com, 2010.

[10] K G Srinivasa, Harish Raddi C S, Mohan Krishna S H, Nidhi Venkatesh, "MeghaOS: Cloud based Operating System and a Framework for Mobile Application Development", World Congress on Information and Communication Technologies, 2011, DOI:10.1109/WICT.2011.6141360.

[11] Fanlong Zhang, Xiaohong Su, Zhigang Sun and Peijun Ma, "The Design and Implementation of Cloud Terminal Operating System Kernel", IEEE fifth International Conference on Advanced Computational Intelligence(ICACI), Nanjing, Jiangsu, China, October 18-20,2012,DOI:10.1109/ICACI.2012.6463125.

[12] H. J. Wang, A. Moshchuk, A. Bush, "Convergence of Desktop and Web Applications on a Multi-Service OS", in HotSec'09 Proceedings of the 4th USENIX conference on Hot topics in security, CA,USA, pp.ll-l1, 2009, DOI: 10.1.1.159.8088.

[13] S. Tang, H. Mai, S. T. King, "Trust and Protection in the Illinois Browser Operating System", in the 9th USENIX conference on Operating systems design and implementation, Vancouver, BC, Canada, pp.I-8, 2010.

[14] H. Hermann, H. Michael, L. Jochen, "The performance of μ -kernel-based systems", in the sixteenth ACM Symposium on Operating Systems Principles, New York, USA, pp.66-77, 1997, DOI:10.1145/269005.266660.

[15] J. N. Herder, H. Bos, B. Gras, "MINIX 3: a highly reliable, self-repairing operating system", in ACM SIGOPS Operating Systems Review, Volume 40, Issue 3, pp. 80-89, July 2006, DOI:10.1145/1151374.1151391.

[16] L4Ka::Pistachio microkerneL (2010). [Online].Available: http://14ka.org/proj ects/pistachio.

[17] GOOGLEINC. (2010). ChromiumOS. [Online]. Available: http://www.chromium.org/chromiumos.ChromeOS/Chromium_OS/chromiumos-design-docs.html.

[18] Google, "Introducing the google chrome-OS,"http://googleblog.blogspot.com/2009/07/introducing-google-chromeos.html [retrieved: 02, 2013].

[19] Lawton, G., "Moving the OS to the Web", Computer Journal, Vol.41, No. 3,p.16-19,2008, DOI:10.1109/MC.2008.94.

[20] R.Want, T. Pering, G. Danneels, M. Kumar, M. Sundar and J. Light, "The personal server: Changing the way we think about ubiquitous computing," in ACM International Conference on biquitous Computing. Goteborg, Sweden: IEEE Computer Society, pp. 194–209, 2002, DOI: 10.1.1. 9. 427.

[21] Microsoft, "Get started using remote desktop with windows xp professionl, "http://www.microsoft.com/windowsxp/using/mobility/getstarted/remo teintro.mspx [retrieved: 02, 2013].

[22] R. Chandra, N. Zeldovich, C. Sapuntzakis, and M.S. Lam, "The Collective: A Cache-Based System Management Architecture," Proc. Second Symp. Networked Systems Design and Implementation (NSDI '05), May 2005.

[23] M. Annamalai, A. Birrell, D. Fetterly, and T. Wobber, "Implementing Portable Desktops: A New Option and Comparisons," Microsoft Research (MSR)-2006-151, Oct. 2006.

[24] [http://www.moka5.com, 2010.

[25] http://www.migosoftware.com/default.php, 2010.

[26] http://www.u3.com, 2009.

[27] http://www.ceedo.com, 2010.

[28] Bickmore, T.W. and Schilit, B.N. Digestor, "Device-Independent Access to the World Wide Web", In Proc.Sixth International

World Wide Web Conference, pp.655-663, 1997, DOI:10.1.1.41.3563