International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:4, 2014

An Adaptive Virtual Desktop Service in Cloud Computing Platform

Shuen-Tai Wang, Hsi-Ya Chang

***Abstract*—**Cloud computing is becoming more and more matured over the last few years and consequently the demands for better cloud services is increasing rapidly. One of the research topics to improve cloud services is the desktop computing in virtualized environment. This paper aims at the development of an adaptive virtual desktop service in cloud computing platform based on our previous research on the virtualization technology. We implement cloud virtual desktop and application software streaming technology that make it possible for providing Virtual Desktop as a Service (VDaaS). Given the development of remote desktop virtualization, it allows shifting the user's desktop from the traditional PC environment to the cloud-enabled environment, which is stored on a remote virtual machine rather than locally. This proposed effort has the potential to positively provide an efficient, resilience and elastic environment for online cloud service. Users no longer need to burden the platform maintenances and drastically reduces the overall cost of hardware and software licenses. Moreover, this flexible remote desktop service represents the next significant step to the mobile workplace, and it lets users access their desktop environments from virtually anywhere.

***Keywords*—**Cloud Computing, Virtualization, Virtual Desktop, VDaaS.

I. Introduction

CLOUD computing [1], [2] is the innovative and leading information technology model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort. The users can request access from a set of web interfaces that manage and monitor a pool of computing resources including machines, network, storage, operating system, application programs and development environments. The software resources sharing in the cloud is also delivered as a service and becomes more popular in recent years. There are many companies such as Amazon and Google have promoted the relative simplicity of the software as service concept. The cloud computing paradigms provide simple and transparent approaches to enable effective sharing and utilization of applications over Internet.

To achieve these goals, adopting virtualization technology [3], [4] in cloud environment becomes a major trend. Virtualization technology acts as a central component that can achieve the purpose of cloud platforms and services, and it is a promising approach to consolidating multiple services onto a smaller number of computing resources. A virtualized server environment allows computing resources to be shared among multiple performance-isolated platforms called virtual machines [5], [6]. A virtual machine is a software implementation of a machine that executes related programs like a physical machine. Each virtual machine includes its own system kernel, OS, supporting libraries and applications. A hypervisor provides a uniform abstraction of the underlying physical machine, and multiple virtual machines can execute simultaneously on a single hypervisor. Decoupling of virtual machine from the underlying physical hardware is able to allow the same virtual machine to be started and run on different physical environments. Thus virtualization is seen as an enabler for cloud computing, allowing the cloud service provider the necessary flexibility to move and allocate the computing resources requested by the user wherever the physical resources are available.

Considerations for an individual user's desktop in such virtualized environment, the virtual desktop has received great interests in virtualization research community. Many authors [7]-[10] have realized the concept of desktop virtualization. In a cloud computing environment, users can utilize SaaS (Software as a Service) subscriptions instead of traditional software licenses. In a traditional computing environment, users need to locally install operating systems and applications under a granted license before use. Software users may be burdened with many complex tasks in terms of software installation, configuration, updating and even troubleshooting when dependency on the host operating system causes compatibility issues. Users access software on demand through the Internet without any updating and maintenance issues. For the system provider there are two alternative methods of making the SaaS software available. One is to develop software based on Web technologies. This not only requires significant work, but may also encounter compatibility problems with the numerous browsers. The second approach is based on desktop virtualization, which separates the presentation and execution of applications. This provides a transparent way to deliver an application based remote virtual desktop.

In this paper, we aim at the adoption of desktop virtualization and cloud technologies. We implement cloud virtual desktop and application software streaming technology that make it possible for providing Virtual Desktop as a Service (VDaaS), which is efficient, resilience and independent of the operating system. The desktop, data and applications used remain on the remote system. We also implemented a sketch of unified web-based interface to make such a service is simple and easy to use for both casual and expert users in any place and any devices.

This work is supported by National Science Council, R.O.C., under the contract number of NSC-102-2221-E-492-022

S.T. Wang and H.Y. Chang are with the National Center for High-Performance Computing, Taiwan, R.O.C. (e-mail: stwang@nchc.org.tw jerry@nchc.org.tw).

The rest of this paper is organized as follows. Section II lists the background. Section III gives a description of software architecture. Section IV gives some details of the implementation. Section V discusses future work and concludes.

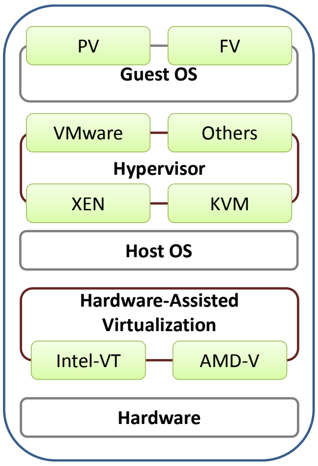

Fig. 1 Virtualization architecture

II. Background

A. Virtualization Architecture

Fig. 1 shows the principal architecture of virtualization. Physical hardware resources were divided as virtual resources of virtualization platforms by the monitors, and those virtual resources were assigned to each virtual machine by different application requirements. Furthermore, the monitor provides each Guest OS a set of virtual platform interfaces that constitute virtual machines, acting as a bridge to connect between hardware and virtual devices. Instructions were delivered to hardware layer from virtual platform, which receives results from monitor of virtual machines. Each virtual platform will run independently, although physical resources were shared. By the way, the monitor has two kinds of model, hypervisor and virtual machine monitor (VMM) [11]. The main distinction between Hypervisor and VMM is that the former monitor runs above hardware layer directly with better performance than VMM, such as Xen [12] and VMware's ESX. On the other hand, Microsoft's Virtual PC and VMware's Workstation adopt VMM as monitor of virtual platforms.

For Guest OS, it includes two main virtualization types [13]: para-virtualization (PV) and full-virtualization (FV), which can be both combined with hardware-assisted virtualization. The Guest OS is simulated by modified kernel of Linux with PV, but related devices are not emulated. Instead, all devices are accessed through light-weight virtual drivers offering better system performance and close to the physical machine. But the drawback is that guest kernels must be upgraded to provide new virtual system calls for the new services and all of guest OS must be compatible with the host OS.

B. Virtual Machine

Virtualization technology is able to apply not only to subsystems but to a complete virtual system. To implement a virtual machine, software developers design a software layer to real machines to support the desired architecture. By providing one or more efficient virtual platforms, virtual machines have extended multi-processing systems of the past decade to become multi-environment systems as well. There are many kinds of virtual machine in the market, but not all virtual machines fit to build virtual platforms, so we choose the Kernel-based Virtual Machine to achieve our purpose. KVM is a open source software with GPL, developing by Qumranet company. KVM provide FV solution for Linux on x86 hardware containing virtualization extensions with Intel-VT or AMD-V, and the kernel component of KVM is encased in mainline Linux OS over version 2.6.20, hence the user can set up the virtualization environment of KVM when installing Linux OS conveniently.

Using KVM, we can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has private virtualized hardware devices, such as network card, disk, graphics adapter, etc. Besides, virtual machine was like a process in queuing system of Linux, and then manager can directly kill any virtual machines by control command.

C. Cloud Service Models

Cloud computing enables hardware and software to be delivered as services, where the term service is used to reflect the fact that they are provided on demand and are paid on a usage. There are three service models of Cloud Computing:

1) Infrastructure as a Service (IaaS): it is a provision model in which an organization outsources the equipment used to support associated operations, such as storage, hardware, servers and network components. This service model allows the cloud to operate during high traffic and demanding situations as resources are dynamically increased as they are needed.
2) Platform as a Service (PaaS): it is the delivery of a virtualized application, computing platform and solution stack as a service. It offers a high-level integrated environment to build, test, and deploy custom applications. All applications and infrastructure are run and managed by the services vendor. Generally, developers will need to accept some restrictions on the type of software they can write in exchange for built-in application scalability.
3) Software as a Service (SaaS): it is a software delivery method that provides access to software and its functions remotely as a Web-based service. Providers offer users access to specific application programs controlled and executed on the provider's infrastructure.

III. Architecture

Table I shows the specification information of our cloud service platform named Formosa 3. Formosa 3 [14] is a 64bits high-performance Beowulf cluster located within Southern Business Unit of the National Center for High Performance Computing (NCHC) [15]. It consists of 76 IBM X3550M3 servers as its compute nodes. This self-made cluster was designed and constructed by the 'HPC Cluster Group' at NCHC for cloud IaaS service and came online in 2012. Each node has two Six-Core Intel Xeon x5660 2.8GHz processors and 48GB of DDR3 registered ECC SDRAM. All nodes were connected on the InfiniBand high speed network and a private subnet with 1000 Mbits/s Gigabit Ethernet. An additional 4 nodes are used as front ends to interface with cluster, and 4 nodes as storage for the user file systems by Parallel File System.

TABLE I
FORMOSA 3 CLOUD CLUSTER SPECIFICATION

CPU	Intel Xeon x5660 six cores 2.8GHz
Hard Disk	80GB SSD
Memory	48GB DDR3 Registered ECC SDRAM
Network	4x QDR 40Gb Infiniband and Gigabit Ethernet
Operating System	CentOS 6.3
Hypervisor	Kernel-based Virtual Machine

The virtual desktop system was developed under the NCHC Formosa 3 cluster platform to provide more application services and reduce the software maintenance costs. The virtual desktop service was built based on the VDI framework, and the main components of VDI framework are: VDI Infrastructure, VDI Desktops and VDI Middleware, which is shown as Fig. 2. VDI Infrastructure is responsibility for creation, deletion or migration of virtual desktop. VDI Middleware is responsibility for connection management and desktop control. Applications and shared data can be installed on the cluster server and its hypervisors, and users can request for some specific applications or data on demand from the server. The requested virtual desktops and applications will be delivered with dynamic streaming technologies to the client devices.

There are two ways to execute the virtual desktop and applications on the client devices:

1) Native mode: The remote control and streaming display software will be installed on the client device to execute the virtual desktops and applications. It is suitable for the homogeneous operating system between the server and clients.
2) Web-based mode: Only need a web browser with Java applet to install on the client device. It is suitable for the heterogeneous operating system between the server and clients.

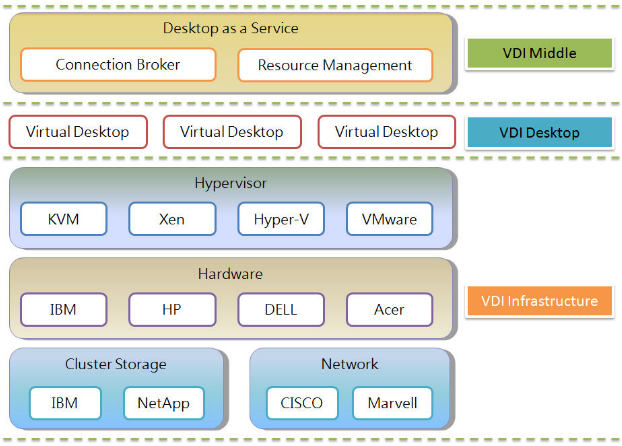

Fig. 2 System architecture

IV. IMPLEMENTATION

The virtual desktop and application sharing system allows more than one person to collaborate or develop on a single document in real-time. Currently virtual desktop is delivered using RFB (Remote Frame Buffer) [16] protocol such as Virtual Network Computing (VNC) and Remote Desktop Protocol (RDP) [17]. These protocols generally provide methods for accessing a remote virtual desktop or application. This system also presents some novel scenarios for application sharing in single or multiple virtual machines. In a single virtual machine, collaboration scenario can be supported which based on a shared desktop, for example, desktop sharing enables the instructor and students to work on the same view in a remote teaching system.

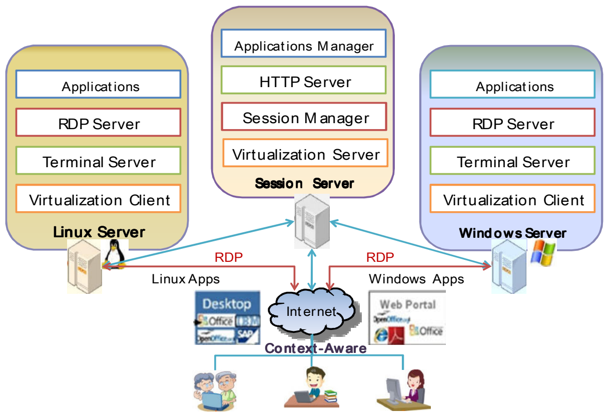

Fig. 3 The virtual desktop and application sharing system

Fig. 3 shows the virtual desktop and application sharing system architecture. This system has a session server to perform the role of virtualization controller, and to manage user's connection and accounting information. The session server also communicates with application servers to deliver the virtual desktop or application streaming to client devices by RDP technology. Besides, the session server adopts the HTTP server to provide the single web-based portal for user login. It also enhances the system security and elasticity by applying the centralized control of applications and files.

Our system is composed of several modules: Desktop Virtualization Manager, Application Virtualization Manager, User Session Manager and Data Synchronization Manager. The flowchart of software modules is shown in Fig. 4. The following is the details of each module:

- Desktop Virtualization Manager: this module provides virtualization of operating system and customization of remote desktop environment. The development of this module is based on the open virtual desktop software, and designed according to the user requirements. It is responsible for allocating computing resources of cluster server and hypervisors dynamically.
- Application Virtualization Manager: this module provides virtualization of applications and software resources. The development of this module is based on the open virtual application software, and designed according to the user requirements. The Full application virtualization also requires a virtualization layer with the operating system. Application virtualization layers replace part of the runtime environment normally provided by the operating system. The layer intercepts all file and registry operations of virtualized applications and transparently redirects them to a virtualized location.
- User Session Manager: this module is responsible for managing user connection sessions and authenticating accounts information. The user session begins when the user accesses the virtual desktop or application and ends when the user quits the virtual desktop or application from the web browser. It also plays a role of bridge between application servers and client devices. It applied the SSH (Secure Shell) and HTTP protocol to provide the single entry website.
- Data Synchronization Manager: this module is responsible for managing the process of establishing consistency among data from cluster servers to client devices and the continuous harmonization of the data over time. Users and project developers can collaborate or develop on a single file without installing any relative application on their own client device. It also backups and shares the user's data to make the system reliable and elastic.

To enable collaborations among multiple virtual machines, the application sharing and migration mechanism will be applied. Through presentation streaming redirection and virtual machine cloning technology, an application can be easily shared or migrated. To realize remote application access in the cloud environment, the RFB protocol is used to transfer the virtual desktop of a remote virtual machine. The RFB protocol works at the buffer frame layer and supports the remote access to graphical user interface, and the mouse or keyboard inputs can be transferred to the remote application, thus achieving a transparent access to the application.

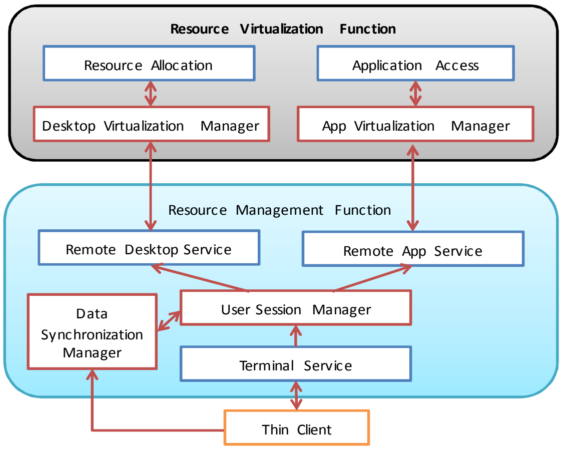

Fig. 4 The flowchart of software modules

While implementing our cloud platform for remote virtual desktop, we came across several issues that have previously not been addressed. For example: for accessing the desktop of the virtual machine users can use Off-the-Shelf tools, ex: VNC and Windows Remote Desktop, etc. Due to the virtual machine may execute on different physical machines every time. This can be troublesome if we provide a fixed public IP address and port for connecting to the user's console of virtual machine. So, we use ip tables and thus setup port forwarding connections to the virtual machine that user launched. Our interface will allocate a mapping port dynamically after user creating virtual machine. After that users can connect to the console with the dedicated IP address of the Server and the port which will be forwarded to the appropriate physical machine which is currently hosting the user's virtual machine. On the other hand, our system also supports both multicast and unicast transmissions. For unicast connections, either UDP or TCP can be used. Since TCP provides reliable communication and flow control, it is more suitable for unicast sessions. Multiple TCP clients sharing a single application may have different bandwidths, so an algorithm which sends the updates at the link speed of each client will be developed. For UDP clients, the system controls the transmission rate because UDP does not provide flow and congestion control. Several simultaneous multicast sessions with different transmission rates can be created at the system. The system can share an application to TCP clients, UDP clients, and several multicast addresses in the same sharing session.

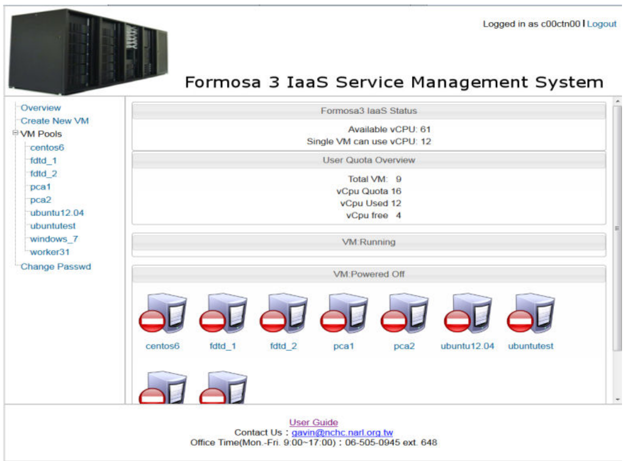

Fig. 5 Cloud service web page

Fig. 5 shows the web page of our cloud service, we create a test account and login the system to operate virtual machines. Related information and status of virtual machines are showed in this main page, including the quota status of cloud platform and users, users can know that how many cores can be used, how many virtual machines are built, how many cores of user's virtual machines, which network ports are allocated for user's virtual machines. And users can manage their virtual machines including create, delete, shutdown, and add network port.

In our cloud service web page, we can create a virtual machine and open a virtual desktop. The desktop and application can be streamed from the virtual machine into an isolation virtual environment on the target device. The desktop view is transmitted to the target device and the application execution at run time is controlled by the user session manager and virtualization layer. Users can operate this virtual environment, so it is as well as a standard physical OS environment.

V. Conclusion

In this paper, we adopt cloud technologies and desktop virtualization to build a virtual desktop and application sharing cloud service which is efficient, resilience and independent of the operating system. We implemented virtual desktop mechanics that make it possible for providing Virtual Desktop as a Service (VDaaS). Given the development of remote desktop virtualization, we can shift the user's desktop from the traditional PC environment to the cloud-enabled environment, which is stored on a remote virtual machine rather than locally. We also implemented a sketch of unified web-based interface to make such a service is simple and easy to use for both casual and expert users. This development has the potential to positively provide an elastic environment for online cloud remote desktop service, and this service represents the next significant step to the mobile workplace. It lets users access their desktop environments from virtually anywhere.

Currently, we support virtual machines that run atop the KVM hypervisor, but plan to add support for VMware, and others in the near future. Also, we plan to optimize the network performance for reducing the communication overhead and adapt some smart management strategies for physical machines to prevent energy waste in cloud computing platform.

References

[1] I. Foster, Y. Zhao, I. Raicu, and S. Lu. "Cloud Computing and Grid Computing 360-Degree Compared," IEEE Grid Computing Environments Workshop, pp. 1-10, 2008.
[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," 2009.
[3] L. Nussbaum, F. Anhalt, O. Mornard and J.-P. Gelas, "Linux-based virtualization for HPC clusters," Linux Symposium, pp. 221-234, July 2009.
[4] G. Goth, "Virtualization: Old Technology Offers Huge New Potential," IEEE Distributed Systems Online, vol. 8, no. 2, 2007.
[5] R. A. Meyer and L. H. Seawright, "A Virtual Machine Time-Sharing System," IBM Systems Journal, vol. 9, no. 3, 1970.
[6] R. P. Goldberg, "Architecture of Virtual Machines, "National Computer Conference Proceedings, AFIPS Press, vol. 42, pp. 309-318, June 1973.
[7] A. Sultana, B. Daimary, M. Chettri, J. Joseph, "Virtualized Remote Web Desktop," IEEE NCETACS National Conference on Emerging Trends and Applications in Computer Science, 2012.
[8] M. Miller, "Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online," Que Publishing, 2009.
[9] H. Lee, "Design for management software of desktop virtualization solutions," IEEE Information and Communication Technology Convergence, ICTC 2010.
[10] L. Yan, "Development and application of desktop virtualization technology," IEEE Communication Software and Networks ICCSN, 2011.
[11] G. J. Popek, and R. P. Goldberg. "Formal Requirements for Virtualizable Third Generation Architectures," Communications of the ACM, vol. 17, pp. 412-421, 1974.
[12] Xen hypervisor, Available at: http://www.xen.org/.
[13] W. Chen, H. Lu, L. Shen, Z. Wang, N. Xiao and Dan Chen, "A Novel Hardware Assisted Full Virtualization Technique," The 9th International Conference for Young Computer Scientists, pp. 1292-1297, Nov. 2008.
[14] NCHC Formosa 3 Cloud Cluster, Available at: http://formosa3.nchc.org.tw/.
[15] NCHC, National Center for High-performance Computing, Available at: http://www.nchc.org.tw.
[16] RFP - Remote Framebuffer Protocol, RFC 6143, Available at: http://www.faqs. org /rfcs/rfc6143.html.
[17] RDP - Remote Desktop Protocol, Available at: http://en.wikipedia.org/wiki/Remote_Desktop_Protocol.