International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 11

ISSN: 2454-4248
99 – 104

A Machine Learning Approach for Desktop and Application Virtualization Design in Cloud Environment

Yu-Ching Lin
National Center for high-Performance Computing
National Applied Research Laboratories
Hsinchu City, Taiwan, R.O.C
e-mail: 1203043@narlabs.org.tw

Abstract**—**In recent years, virtual desktop and virtual application is the important research topic for virtualization of cloud computing. Virtualization provides many benefits by using virtual machine software, including we can efficiently deploy and manage all of virtual system resources, and it offers the ability of high reliability, high elasticity and customization. In order to share the system and software resources, related basic knowledge show in our paper about the virtualization technology of desktop and application, and we proposed the virtual desktop and application services to offer an efficient and elastic service for cloud platform. A machine learning approach is also applied to manage resource allocation. It implements the VDaaS (Virtual Desktop as a Service) and VAaaS (Virtual Application as a Service) by developing the sharing technology for virtual desktop and virtual application with the cloud platform.

Keywords*-virtualization; desktop; application; machine learning*

*****

I. INTRODUCTION

The Cloud computing is a very popular Information Technology model for enabling fast, convenient, on-demand network access to a shared pool of related computing resources, which include servers, networks, storage, software, data, applications and services. Those computing resources can be rapidly provisioned and released with minimal management effort or service provider interaction. There are many benefits with cloud technology including high scalability, high reliability, low cost, simple management, facilitation, on-demand, customization and pay-per-use. Cloud service is the access to computers and their functionality through three difference deployment models, including public, private, and hybrid. Users of the cloud request this access from a set of web service interfaces that manage and monitor a pool of computing resources base on the virtualization technology including virtual machines, network devices, storage, operating system, application programs and development environments [1].

With the growing adoption of virtualized datacenters and cloud hosting services, the allocation and sizing of resources such as CPU, memory, and I/O bandwidth for virtual machines (VMs) is becoming increasingly important. Accurate performance modeling of an application would help users in better VM sizing, thus reducing costs. It can also benefit cloud service providers who can offer a new charging model based on the VMs' performance instead of their configured sizes. Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service [2, 3, 4].

When granted service requests from users, a part of the computing resources in the resource pool is dedicated to the requesting user until those resources are released. Users are allowed to use base on three types of the cloud service models. The cloud service models show below:

- Infrastructure as a Service (IaaS): The IaaS services are self-service models for accessing, monitoring, and managing remote datacenter infrastructures, 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.
- Platform as a Service (PaaS): The PaaS services are used for applications, and other development, while providing cloud components to software. PaaS makes the development, testing, and deployment of applications quick, simple, and cost-effective. All applications and infrastructure are run and managed by the services vendor.
- Software as a Service (SaaS): The SaaS services represent 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.

Virtualization is the act of isolating or unbinding one computing resource from others. Or to put it in another way: the process of decoupling layers of IT functions so that configurations of the layers become more independent of each other. As a result, virtualization masks the specific nature of IT resources from users. A user sees the function, not the resource that delivers the function [5, 6, 7].

Application virtualization is an essential and critical component for all desktop delivery solutions. The term "Application Streaming and Virtualization" is often used by customers, although it's rarely fully understood. To understand the true meaning it's best to break down into constituent parts:

- Application: These are the end-user focused Windows and web programs, executed on a Windows Operating System Platform.
- Streaming: The process of transporting the application specific data/content as quickly as possible to the end-point. The application is quick-up-and-running where ultimately the required resources to run and use the application is being delivered to the user while the remaining data is transferred in the background.
- Virtualization: The software layer that improves portability, manageability and compatibility of applications by encapsulating them from the underlying operating system on which they are executed.

ISSN: 2454-4248
99 – 104

With application deployment, the applications are installed on the execution platform. The execution platform could be a local desktop or laptop, a server hosted virtual desktop or a remote desktop server session host. When speaking of application delivery in the context of application virtualization, the applications are no longer installed, but they are made almost instantly available and executed on the execution platform. The execution platform is not altered. Application virtualization enables fast application delivery in a central and local environment.

Besides, the Virtualization Technology acts a very important role that can achieve the purpose of cloud platforms and services, and itis the ability to run many virtual machines on top of a hypervisor. A virtual machine (VM) is a software implementation of a machine that executes related programs like a physical machine. Each VM includes its own system kernel, OS, supporting libraries and applications. A hypervisor provides a uniform abstraction of the underlying physical machine, and multiple VMs can execute simultaneously on a single hypervisor. The decoupling of VM from the underlying physical hardware is able to allow the same VM to be started and ran 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 related computing resources requested by the user wherever the physical resources are available [8, 9, 10].

In this paper, the main goal is to develop a virtual desktop and application sharing system which is scalable, efficient, and independent of the operating system. The virtual desktop and application sharing system provides an open architecture to deliver desktops and applications with the heterogeneous operating systems to any client device of users. Users can easily use the operating system environment or applications to develop software projects in the cloud through the web browsers over Internet.

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. Users may be burdened with many complex tasks in terms of software installation, configuration, updating and even troubleshooting issues. With the cloud computing concept, users may access software on demand through the Internet without any installation and maintenance issues. For the system providers, 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 the remote virtual desktop and applications, which separates the presentation and execution of operating systems and software. This provides a transparent way to deliver software resources based on the virtualization technologies. By utilizing the virtualization technologies, all cluster servers and the software which are maintained by the cloud providers and the users just pay as they go. Cloud service providers provide the resources sharing system for accessing virtual desktop and applications, so users develop their programs and execute in the cloud environment provided by the cloud service providers.

The system technology is discussed in section II, and system design and implementation of the system is discussed in section III. Conclusion of this paper will be in section IV.

II. SYSTEM TECHNOLOGY

The key technology of the virtual desktop and application sharing system is virtualizing and sharing of desktops and applications. The virtualization software is installed with the cluster server and its virtual machines. Each virtual desktop and application shares the computing resources of the cluster server. The cluster server is one of the important infrastructures in the cloud computing technology, and it is one of the important resources in the Internet. The main purpose of the cluster server is to formulate the responses which are sending by the clients, it also performs the several load balancing activities while the tasks access to make the system reliable.

The virtual desktop and application sharing system allows more than one person to collaborate or develop on a single document at the same time. Currently virtual desktops are delivered using RFB (remote framebuffer) protocols such as VNC (Virtual Network Computing) and RDP (Remote Desktop Protocol) [11, 12]. These protocols generally provide methods for accessing a remote virtual desktop or application, and users can login to a system server and work on the desktop. The virtual desktop and application sharing system is also presenting some innovative scenarios for application sharing in single or multiple virtual machines. In a single virtual machine, collaboration scenarios can be supported 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.

A. *Virtualization of Application*

Application virtualization is layered on top of virtualization technologies. Application virtualization is the separation of an installation of an application from the client computer that is accessing it. Application virtualization can be divided into two types: remote and streaming. Remote applications run on a server. End users view and interact with their applications over a network via a remote display protocol. The remote applications can be completely integrated with the user's desktop so that they appear and behave the same as local applications. The remote applications can be shared with other users, or the application can be running on its own operating system instance on the server. With streaming applications, the virtualized application is executed on the end user's local computer. When an application is requested, components are downloaded to the local computer on demand. Once completely downloaded, a streamed application can work without a network connection. Both forms of application virtualization have benefits from centralized management. Applications can be installed, patched, and upgraded once for an entire environment. Application virtualization typically is combined with application streaming, so the software is not installed in the usual sense. Application virtualization allows software to be run without being installed on the machine and without interacting directly with the original operating system for which it was designed. Application streaming is an on-demand software distribution system. With application streaming, users receive only applications that they request from a master server. In addition, clients are given only the portion of the application that they need to launch the software and use its basic functions [13, 14].

The virtual application sharing is different from the virtual desktop sharing. In virtual desktop sharing, a server distributes any screen update. In virtual application sharing, the server distributes screen updates if they belong to the shared application's windows. Client users receive screen updates from the server and send keyboard and mouse events to the server. Application virtualization is also a software technology that encapsulates applications from the underlying operating system on which it is executed. With application virtualization, users can store and manage applications centrally, integrate with third party computer lifecycle management and software distribution systems. Application virtualization also enhances organizations' ability to control access to applications, track usage of virtual applications, and quickly test, deploy and update applications.

Full application virtualization requires a virtualization layer that replaces part of the runtime environment normally provided by the operating system. The application virtualization layers are used to substitute part of the runtime environment which is provided by the operating system. The layer acquires registry operations of virtualized applications and transparently redirects them to a virtualized location. It makes the real application remaining to access the virtual resources. The pieces of the application's code, data, and settings are delivered when first used, and the application streaming are usually delivered over a protocol such as HTTP (Hypertext Transfer Protocol), CIFS (Common Internet File System) or RTSP (Real Time Streaming Protocol) [15, 16]. Deploying and maintaining desktop and application level virtualization poses many challenges, but there are opportunities for improvements at the application and server layers to achieve the high-level of computing performance and scalability [17, 18]. The application is executed inside the Virtual Environment, (sometimes called bubble or sandbox) and behaves as if it is running alone in the Operating System. So the underlying Operating System is protected, since the application virtualization prevents changes to system components. Applications can use the hardware and software components that are installed and available inside the Operating System. While most application virtualization technologies today provide an adequate level of isolation between applications, thus preventing app-to-app conflicts, very few provide full OS isolation and are able to prevent app-to-OS conflicts.

Application virtualization decouples applications from underlying infrastructure by creating a virtual file system and virtual registry to provide the file and registry resources to the application at runtime. The 1st application virtualization generation approach was to create a virtual full replica of the file system and registry and all the APIs that applications use to interface with them. However, this caused two major issues. Firstly, local applications had no way to access the virtual file system and registry so many application integrations are unable to work. Secondly, it meant that all of the APIs that applications use to communicate with the file system and registry had to be re-implemented in the application virtualization technology.

B. Machine Learning Approaches

Machine learning techniques have been extensively studied for performance analysis and troubleshooting. The CARVE project employs simple regression analysis to predict the performance impact of memory allocation. Wood et al. use regression to map a resource usage profile obtained from a physical system to one that can be used on a virtualized system. However, the accuracy of regression analysis has been shown to be poor when used for modeling the performance of virtualized applications. The Tree-Augmented Bayesian Networks is used to identify system metrics attributable to SLA violations. The models enable an administrator to forecast whether certain values for specific system parameters are indictors of application failures or performance violations. In subsequent work, the authors used Bayesian networks to construct signatures of the performance problems based on performance statistics and clustering similar signatures to support searching for previously recorded instances of observed performance problems.

The VCONF project has studied using reinforcement learning combined with ANN (Artificial Neural Networks) to automatically tune the CPU and memory configurations of a VM in order to achieve good performance for its hosted application [19]. The most solutions are specifically targeted for the CPU resource. In addition to CPU, we address memory and I/O contention explicitly. To address such questions, ANN models were proposed in the previous work [20]. However, subsequent investigations revealed several drawbacks. First, we observed that the parameter to capture I/O contention in shared storage platform can lead to arbitrary inaccuracy in the model. Second, we observed that the proposed approach of constructing a single model encompassing the entire parameter space in a multi-dimensional model was also severely deficient. We explore the use of both ANN and SVM approaches to machine learning for performance modeling. Although SVMs are generally applied as a powerful classification technique, SVM-based regression (SVR) is gaining popularity in systems data modeling.

An ANN has multiple layers of neurons which include at least one input layer and one output layer and a tunable number of hidden layers, each with a tunable number of hidden neurons associated with it. The number of hidden layers and hidden neurons at each layer play an integral role in determining the accuracy of an ANN model. However, choosing the number of hidden neurons at each hidden layer is not straightforward. If the number of hidden neurons is too small, the model may not assign enough importance to the input parameters and hence suffer from under-fitting. Conversely, if this number is too large, the model may amplify the importance of input parameters and suffer from over-fitting.

We evaluated an automated incremental approach of determining the hidden layers and hidden neurons in the training process. It starts with one hidden layer and three hidden neurons at each hidden layer to accommodate the three modeling parameters. If adding more neurons or layers does not lead to improved accuracy, we pick the least number of neurons which provided the best accuracy.

We propose to use advanced machine learning methods to model the relationship between the resource allocation to a virtualized application and its performance using a limited amount of training data. Such a model can then be used to predict the resource need of an application to meet its performance target. One of the questions in this approach is when and how the model is built.

Since our approach requires collecting application performance data for a wide range of resource allocation choices, it is difficult to build the model quickly based only on observations from production runs. One option is to have a staging area where a customer can deploy the application and run a sample workload against various resource allocation configurations to facilitate modeling. The modeling techniques that we propose can complement and enhance such a system which used simple linear interpolation to predict performance results for unavailable allocations.

ISSN: 2454-4248
99 – 104

III. SYSTEM DESIGN AND IMPLEMENTATION

The virtual desktop and application sharing system can be developed under the personal computer or cluster servers to provide more application services and reduce the software maintenance costs. The virtual desktop service is 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 Figure 1. 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.

Virtual desktop infrastructure is the practice of hosting a desktop operating system within a virtual machine running on a centralized server. VDI is a variation on the client/server computing model, sometimes referred to as server-based computing.

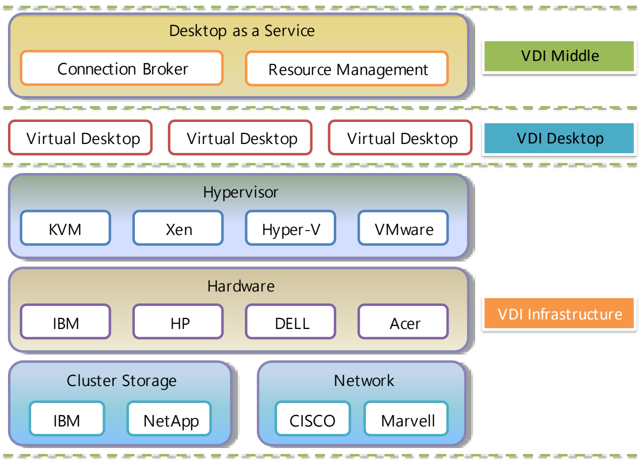

Figure 1. Virtual Desktop Infrastructure Framework

The virtual desktop and application sharing system architecture is shown as Figure 2. This system has a session server to perform in the role of virtualization controller, and to manage user’s connections and accountings 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 uses 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 on the server cluster.

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

- 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.
- 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.

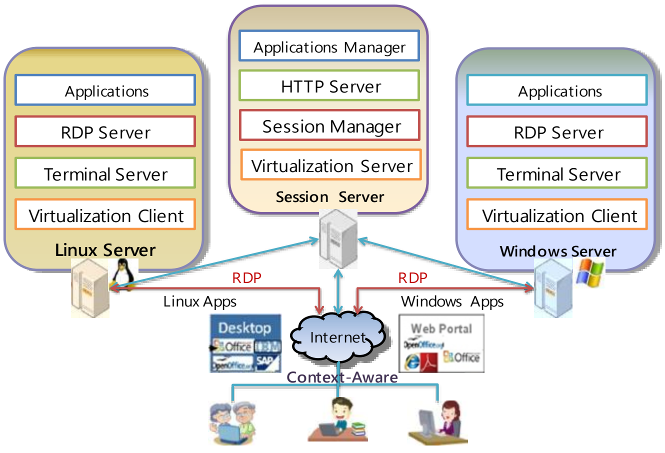

Figure 2. The virtual desktop and application sharing system architecture

The virtual desktop and application sharing system architecture has a session server to perform in the role of virtualization controller, and to manage user’s connections and accountings information. The session server can be implemented with Linux or Windows operating system, and it maintains the virtualization, session and application manager functions. It is responsible for virtualizing the desktop and applications and sends them to the proper sessions for the application servers. 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 uses 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 on the server cluster.

When a user requests the application which installed on the Linux application server, the session server will manage the user account and its session, and communicate with Linux application servers. When the Linux application server receives the request from session server, it will start RDP service and make the application virtualize to streaming data, and then delivers the data for the client user. The session server can also apply the context-aware service to fit the requirements of user’s client device and the network bandwidth, which can enhance qualities of the virtualization services.

The limitations of standard regression analysis techniques for modeling virtualized applications are due to their inability to capture the complex behavior [21]. Specifically, the non-linear dependence of performance on resource levels and the complex influence of contention cannot be represented using simple functions. The study also motivated and justified the choice of ANN-based modeling for virtualized applications. The simple application of ANN-based technique for modeling can produce large modeling errors when used for complex applications and when possible resource allocations span a wider range (as typical in cloud data centers) than those explored in that study.

The use of another popular machine learning model, Support Vector Machine (SVM) which has gained more popularity recently and fined that it has similar limitations as ANN when used directly for modeling. The root cause of why ANN and SVM techniques are still not sufficient and propose improved use of these tools to substantially increase modeling accuracy.

During the training process, a machine learning model gradually tunes its internal network by utilizing the training data set. The accuracy of any model is contingent upon selection of a proper training data set and is evaluated using a different testing data set. Briefly, the training starts with a boot-strapping phase which requires system administrators to identify the best case and worst case resource allocation considered feasible across each resource dimension (CPU limit, memory limit, and virtual disk I/O latency) for the workload on the target hardware. The input parameter set is then chosen by first including these boundary allocation values and selecting additional values obtained by equally dividing the range between the lowest and highest values across each resource dimension. This input parameter set and the corresponding output values (obtained by running the workload on the target system) are chosen as the initial training data set.

After this initial training, the modeling accuracy with the initial training data set is measured by predicting for the testing data set. If satisfactory accuracy (defined as an administrator chosen bound on prediction error) is achieved, the training process concludes. Otherwise, additional allocation values are computed by preferentially varying highly correlated input parameters (based on the correlation coefficient calculated using any statistical tool) by further subdividing the allocation range with the goal of populating the training set with allocation values that represent the output parameter range more uniformly.

The virtual desktop and application sharing system is composed of several modules: Desktop Virtualization Manager, Application Virtualization Manager, User Session Manager and Data Synchronization Manager. The software modules of the system are shown as Figure 3. The functions of each module are described as follows:

1) 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.

2) 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.

3) 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 (Hyper Text Transfer Protocol) protocol to provide the single entry website.

4) 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.

5) Machine Learning Manager: This module provides advanced machine learning methods to model the relationship between the resource allocation to a virtualized application and its performance using a limited amount of training data. Such a model can then be used to predict the resource need of an application to meet its performance target.

Accurate modeling of an application's performance is very useful both for users and cloud service providers. However, the non-linear relationship of resource allocations to application performance, the interplay of resource usage across different types of resources, and resource contention in a virtualized environment makes it very difficult to create such models. We identified three VM resource control parameters as sufficient inputs for creating performance models. We also evaluated the effectiveness of ANN and SVM techniques for modeling and demonstrated that these are superior to conventional regression-based approaches. In order to further increase the accuracy of these techniques, a sub-modeling technique that substantially reduced prediction errors is also presented. Cloud users and service providers can benefit from the ability to create accurate performance models for their virtualized workloads and utilize such models for accurate VM sizing and placement.

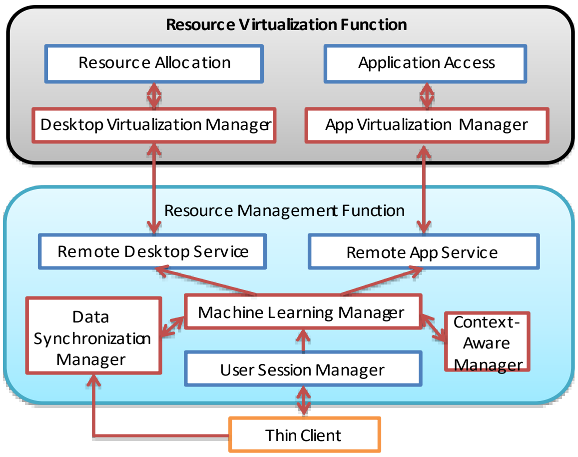

Figure 3. The software modules of the virtual desktop and application sharing system

To enable collaborations among multiple virtual machines, an 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 a cloud, RFB protocol is used to transfer the virtual desktop or application of a remote virtual machine. The RFB protocol works at the buffer frame layer and supports the remote access to graphical user interfaces, and the mouse or keyboard inputs can be transferred to the remote application, thus achieving a transparent access to the applications. In such a presentation streaming based software delivery mode, when a client wants to migrate or share an application to another client, the presentation streaming of this application should be redirected and the corresponding virtual machine will be cloned in case of application sharing.

ISSN: 2454-4248
99 – 104

The virtual desktop and application sharing system 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.

The desktop and application virtualization technology enables operating system environments and applications to be streamed from a centralized location into an isolation environment on the target device where they will execute. The desktop and application files, configuration, and settings are transmitted to the target device and the application execution at run time is controlled by the user session manager and virtualization layer. The virtual desktop and application sharing system enables collaborative works, and provides an efficient, reliable and elastic service platform for cloud-based computing environment.

IV. CONCLUSIONS

In recent years, virtualization technologies have attracted attention to solve the problem of the decreasing efficiency of hardware and software. In order to share the software resources, virtualization technology plays an important role in the cloud computing environment. The development of the virtual desktop and application sharing system will integrate the relative technologies of IaaS, PaaS and SaaS with Cloud computing to provide the innovative application services and realize the VDaaS(Virtual Desktop as a Service) and VAaaS(Virtual Application as a Service). Users no longer need to burden the costs of software license and maintenance, and they only need some simple operations to get the various software resources on demand and pay per use. A machine learning approach is also applied to manage resource allocation. We can provide more innovative application services of Cloud computing for users and make the cluster server as a high economic benefits platform by developing the virtual desktop and application sharing technology. Furthermore, the relative technologies will also push forward the development of Cloud computing field.

REFERENCES

[1] M. A. Vouk, “Cloud computing – issues, research and implementations,” Proceedings of the ITI 2008 30th Int. Conf. on Information Technology Interfaces, pp. 31-40, June 2008.

[2] Wei Chen, Hongyi Lu, Li Shen, Zhiying Wang, Nong Xiao and Dan Chen, “A novel hardware assisted full virtualization technique,” The 9th International Conference for Young Computer Scientists, pp. 1292-1297, Nov. 2008.

[3] Aihua Liang, Zhuo Liu, Limin Xiao, Li Ruan, “An integrated resource allocation algorithm for CPU intensive and I/O intensive jobs in multi-core cluster,” IEEE ICFCC 2010.

[4] V. Talwar, B. Agarwalla, S. Basu, R. Kumar, and K. Nahrstedt, "Resource allocation for remote desktop sessions in utility grids," Concurrency and Computation: Practice and Experience, vol. 18, no. 6, pp. 667-684, 2006.

[5] P. Simoens, P. Praet, B. Vankeirbilck, J. De Wachter, L. Deboosere, F. De Turck, B. Dhoedt, and P. Demeester, “Design and implementation of a hybrid remote display protocol to optimize multimedia experience on thin client devices,” in Proc. Telecommunication Networks and Applications Conference, Adelaide, SA., 2008, pp. 391-396.

[6] Amina Sultana, Bittu Daimary, Mahesh Chettri, and Joby Joseph, “Virtualized remote web desktop,” IEEE NCETACS National Conference on Emerging Trends and Applications in Computer Science, 2012.

[7] Hyungjik Lee, “Design for management software of desktop virtualization solutions,” IEEE Information and Communication Technology Convergence, ICTC 2010.

[8] K. Miller and M. Pegah, “Virtualization: virtually at the desktop,” Proceedings of the 35th annual ACM SIGUCCS conference on User services, Orlando, Florida, USA: ACM, 2007, pp. 255-260.

[9] Li Yan, “Development and application of desktop virtualization technology,” IEEE Communication Software and Networks ICCSN, 2011.

[10] Hai Jin, “Desktop virtualization: techniques and applications,” Pervasive Computing and Applications, ICPCA 2008.

[11] J. Ott, S. Wenger, N. Sato, C. Burmeister, and J. Rey, “Extended RTP profile for real-time transport control protocol (RTCP)-based feedback (RTP/AVPF),” RFC 4585, 2006.

[12] G. Wallace and K. Li, “Virtually shared displays and user input devices,” In 2007 USENIX Annual Technical Conference.

[13] Qian Hao , “Context modeling and reasoning based on certainty factor,” Computational Intelligence and Industrial Applications, PACIIA 2009.

[14] Alexander Gunka, Stepan Seycek, Harald Kühn, “Moving an application to the cloud: an evolutionary approach”, Proceedings of the 2013 international workshop on Multi-cloud applications and federated clouds, April 2013.

[15] Yongseong Cho, Jongsun Choi, Jaeyoung Choi, “An integrated management system of virtual resources based on virtualization API and data distribution service”, Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference, August 2013.

[16] G. Lewis, S. M. Hasan, V. N. Alexandrov, and M. T. Dove. “Facilitating collaboration and application sharing with MAST and the access grid development infrastructures,” In E-SCIENCE ’06, 2006.

[17] S. Xia, D. Sun, C. Sun, D. Chen, and H. Shen, “Leveraging single-user applications for multi-user collaboration: the coword approach,” In CSCW ’04: Proceedings of the 2004 ACM conference on Computer supported cooperative work.

[18] Kudryavtsev, A., Koshelev, V., Avetisyan, “Modern HPC cluster virtualization using KVM and palacios”, High Performance Computing (HiPC), Inter. Conf, Dec.2012.

[19] J. Rao, X. Bu, C.-Z. Xu, L. Y. Wang, and G. G. Yin. VCONF: a reinforcement learning approach to virtual machines auto-configuration. In ICAC, pages 137–146. ACM, 2009.

[20] S. Kundu, R. Rangaswami, K. Dutta, and M. Zhao. Application Performance Modeling in a Virtualized Environment. In Proc. of IEEE HPCA, January 2010.

[21] S. Kundu, R. Rangaswami, K. Dutta, and M. Zhao. Application Performance Modeling in a Virtualized Environment. In Proc. of IEEE High Performance Computer Architecture, January 2010.