Server Based Desktop Virtualization with Mobile Thin Clients

Prof. Sangita Chaudhari
Email: sangita123sp@rediffmail.com

Amod N. Narvekar
Email: amod.narvekar@rediffmail.com

Abhishek V. Potnis
Email: abhishekvpotnis@gmail.com

Pratik J. Patil
Email: pratikpatil.pp @gmail.com

Department of Computer Engineering, Vidyavardhini's College Of Engineering And Technology,
Mumbai University, India

***ABSTRACT* - Virtualization emerged in the 1960s as a practice to optimize the use of expensive computing hardware. Desktop virtualization targets to reduce the complexity associated with deployment and maintenance of client devices, which ultimately helps IT Companies to reduce desktop management costs. The use of thin clients also helps reduce the costs at the client side.**

In Server-based Desktop Virtualization, the users physically work on thin clients, but the operational environment that they interact with is actually running on a remote Server. The user input is transmitted across the network to the remote Server, and the user interface – i.e., the virtual desktop – is presented back through the network to the end-user. In this paper, we propose a server based desktop virtualization system, in which mobile phones can be used as thin clients and a desktop computer acts as the server. All the processing is done centrally on the server itself. Nothing is executed or persistent at the client side. The proposed system will greatly improve the QoS of users and efficiently allocate resources among all clients using appropriate algorithms for the same.

Keywords – Desktop Virtualization, mobile thin clients, cloud computing.

INTRODUCTION

A type of virtualization is actually rapidly growing faster than either server or OS virtualization, both in market and in importance – desktop virtualization [3]. The primitive concept of desktop virtualization is based on application execution from server to remote client. Virtualization extracts the physical physiognomies of computing from the way in which other systems, applications, or end-users interact with those resources. It permits individuals to use computing competences – from a solo application to an complete operating system – without being tangled to the explicit physical hardware or other resources that are supporting to those capabilities.

Server-centered desktop virtualization [10] abstracts physical desktops from how end-users work together with them. The users actually works with their thin client, but the computing environment that they interact with is actually running on a single system which may be a remote one– typically a data center server. Any user input (keystrokes, mouse clicks, etc.) is sent across the network to that system, and the user interface is presented back across the network to the end-user.

Unlike client-server computing [9], application virtualization and application streaming – which all focus on delivering individual applications – desktop virtualization provides a complete operating environment, including the operating system, applications, and data. In addition, unlike systems that provide virtualization on desktop systems, server-based desktop virtualization removes the dependency on specific local PC configurations. In addition, unlike systems that provide virtualization on desktop systems, server-based desktop virtualization eliminates the dependency on specific local PC configurations.

Desktop virtualization could imbibe the use of several different virtualization technologies including the following:

Access virtualization - Individuals access applications running in virtual machines on servers that are situated in the datacenter. Occasionally this access virtualization [10] technology is known as "presentation services." Individual members use a PC running special software or a very lightly constituted computing device known as a "thin client" to access their applications and data.

Application virtualization - Another approach to desktop virtualization is for individuals to access applications that have been summarized and streamed, in whole or in part, down to their local computing device. Once the individual has completed the task, the application could be automatically removed from the local machine and made available for reuse on another machine or it could remain on the local machine. Streaming applications often require broadband network connections.

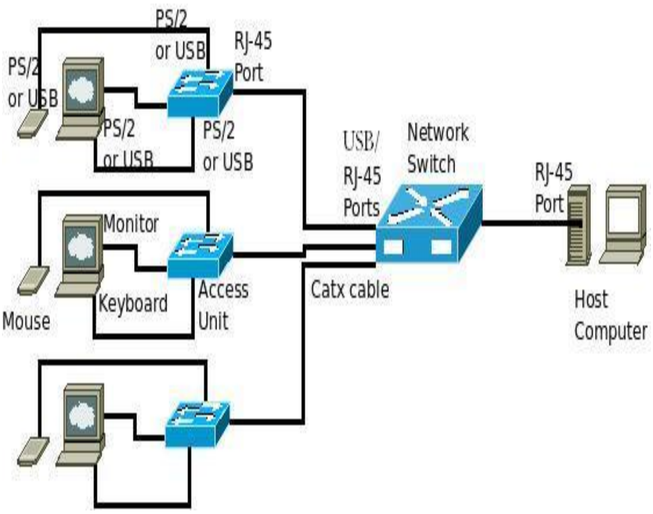

Fig.1. Working of Desktop Virtualization

In desktop virtualization, processing is done using powerful and well-equipped servers in the data center and only the image or the user interface is sent to the desktop or thin client computers. The another way of virtualizing desktops, by sharing or distributing the processor/memory of a single desktop to create multiple desktops by just adding monitors, keyboard, mouse and low cost access devices for each user. The architecture for creating such shared desktops is shown in the above diagram. Each virtualized desktop has an access unit which can be used instead of CPU that connects to the network switch using RJ-45 or USB ports, on one side and keyboard/mouse/monitor using common interfaces like PS/2, USB, and VGA on the other side. On the right hand side, the network switch connects to a host computer (whose hardware is being shared to create multiple virtual computers). Desktop virtualization software is required on the host computer to synchronize the receiving of inputs/ sending of display images to the individual virtual desktops. So, when this set up is complete, any user working on any of the virtual desktop thinks that they are working on a dedicated computer while in actuality, the data is processed on the CPU of the host computer and only the output is shared to the virtual computers. And for this purpose, some specialized protocol might be employed to accelerate the display output. The files of the virtual desktops are stored in the host desktop.

Basically this works on the principle that the desktop computers available today with multi-core processors/ high capacity of RAM etc, are quite powerful and its huge amount of processor/ memory capacity is never completely utilized by a single desktop. So, that excess capacity can be shared – either with one more desktop user/ few more desktop users – depending on the load/ type of applications.

RELATED RESEARCH

Infrastructure virtualization [10] is a powerful trend in the industry today, but the concept of virtualization has been around for many years. Virtualization technology has been pursued by vendors as a framework to divide the resources of a computer into multiple execution environments using a variety of hardware and software abstraction techniques, resulting in the separation of the logical interface from the physical machine.

There exist several protocols used to access remote desktops enabling desktop virtualization. They all share the same concept of relaying the keyboard and mouse events from the client to the server .The server provide clients only with screen region. Multiple ongoing efforts have been made to enrich these protocols with capabilities more suitable to video transmission and handling of peripheral devices on the client systems.

Citrix Systems [1] continues to be the market leader by providing seamless desktop virtualization solutions that provide remote access to centralized business resources.
Intel IT [5] is investigating dynamic virtual client technology, which uses containerized software appliances to abstract the OS; applications; corporate and personal data and workspaces; and user-specific settings. With this model, users can access their applications and information from any device. XenApp [4] also provides application streaming, and alongside it, the Fort Lauderdale-based software manufacturer offers a solution that works with desktop images in the data center.

A DESKTOP VIRTUALIZATION SYSTEM

In this section, we present the design of the proposed Desktop Virtualization System.

SYSTEM OVERVIEW

The proposed System comprises of Mobile Devices being the thin clients and the Desktop PC as a Server. The Mobile Devices may use technologies like Bluetooth or Wi-Fi to communicate with the Server. The general mechanism of the system can be explained as - The mobile (thin client) will send the request to execute a particular application that would otherwise not run on the mobile device, to the server.

The server will respond by executing that application at its own end and only sending the user-interface to the mobile (thin client). At the client-side, the user may give any input, which is transmitted back to the server. The server will process the user input and send the resulting UI back to the client. In this manner, the proposed system will help clients to also execute the applications that otherwise would be impossible to execute on their devices.

SYSTEM ARCHITECTURE

The proposed System comprises of four layers as shown in the figure.

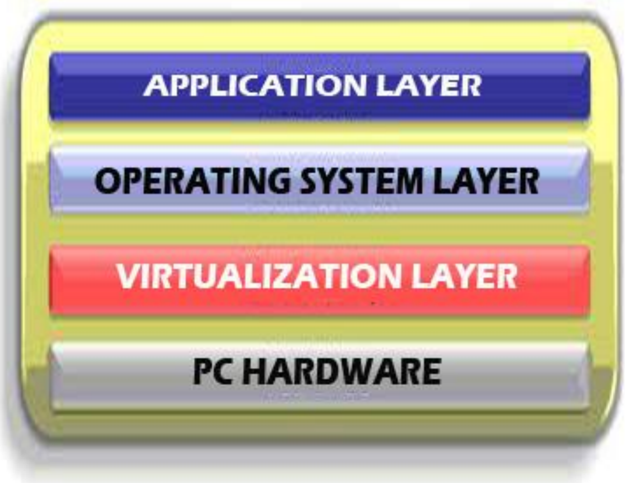

Fig.2 Layers of Desktop Virtualization

PC HARDWARE:

When this approach is applied to desktops, it is clear that the flexible computing components that form the basis of flexible desktops and the user's operating environment can be dynamically delivered to any end-user device, maintaining a consistent delivery experience regardless of endpoint scenario. Conversely, a *purpose-built* desktop can be delivered only with the applications, settings, and services currently required by the user.

VIRTUALIZATION LAYER:

With purpose-built desktops, if all a user needs for a session is access to an e-mail application, they do not need to wait for the desktop indexing service to initialize. When combined with a centralized application and data infrastructure, resources are conserved by delivering only the applications that are required. Further, a defined set of standard builds can be secured and delivered, allowing for consistency in the implementation of security measures.

APPLICATION LAYER

Application Delivery can also spread or mitigate the risks inherent in complex environments where applications coexist. Overall availability can be increased by spreading services across the entire logical datacenter, or in response to an individual outage.

OPERATING SYSTEM LAYER

More efficient utilization of hardware and other resources can be achieved when resources are not dedicated to a specific service. Over-utilized services can be distributed to be shared by under-utilized platforms.

This process of decoupling is so fundamental to best practice application delivery that there are very few exception cases. Exceptions may exist when a load is fixed and predictable, or when privacy or other legal issues arise; or if the layering created by decoupling creates an overhead that reduces the benefits significantly.

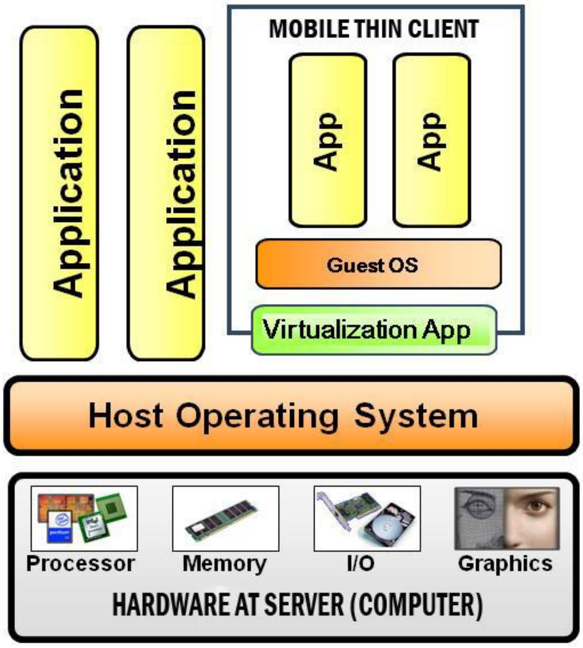

Fig.3 System Architecture

The above four layers are diagrammatically explained in the above figure. In order to make this System a reality, following criteria should be fulfilled.

CLIENT REQUIREMENTS:

Providing users with the flexibility to work anywhere and everywhere but in a consistent and well-coordinated manner. Key to this is effective access to the applications and data that are important to them from both Windows and non-Windows devices regardless of form factor Enhancing business continuity by minimizing the impact of malfunctioning PCs, power outages, natural disasters, and other disruptive events. Key to this is reducing the user's dependency on a specific work location and/or specific items (or types) of equipment to perform their duties effectively.

SERVER REQUIREMENTS:

Effective desktop management enabled by desktop virtualization, including the ability to centrally maintain and update individual PCs, has important implications for information security. Industry analyst firm Gartner Inc. estimates that 90 percent of successful attacks occur against previously known vulnerabilities where a patch or secure configuration standard was already available.

By centralizing desktops, applications and data, desktop virtualization makes it possible for IT to deploy application and security updates and patches in a more timely and consistent manner, significantly reducing vulnerabilities and potential exploitations. The ability to instantly de-provision access to information resources for any worker, regardless of the location of their endpoint, is especially valuable as organizations rely on more third-party partners, vendors and workers, and delivers considerable protection against threats such as employee malfeasance.

NETWORK REQUIREMENTS:

Bandwidth: Different VDI technologies have different traffic patterns and network bandwidth requirements. In general, server-based desktop and virtualization have long-term variable bit rate (VBR) traffic patterns, mainly because of the characteristics of the RDP and ICA protocols, which use compression techniques. The maximum bandwidth requirement can vary greatly depending on user activities, which directly affect the complexity of the desktop display. For video applications, server-side image rendering can greatly increase the bandwidth requirement to several hundred Mbps. For client-based desktop virtualization or application streams, bursty bulk data transfers may occur when the OS and applications are streamed to the client.

Given these bandwidth considerations, ample network bandwidth should be allocated to handle various scenarios. Ethernet 10/100/1000-Mbps host-facing connectivity would provide bandwidth headroom. Correspondingly, bandwidth headroom should be planed in the campus distribution and core layers.

Security: Server-based desktop and application virtualization technologies provide good data protection. In addition, dedicated thin-client devices in general have less exposure to security vulnerabilities due to their simplicity. Client-based desktop virtualization and application streaming technologies with standard PCs have security requirements similar to those for today's PC environments. When planning for network security services, the overall workspace environment needs to be considered. In a mixed desktop environment, proper security services are required to protect the client and network infrastructure.

SERVER-SIDE SCHEDULING ALGORITHM

An identifier is allocated to all clients for unique identification. A queue is maintained to store the order in which clients request then server. A combination of First Come First Serve and Round Robin Algorithm is used to serve the multiple clients at the server-side.

ALGORITHM:

1. Start

2. Listen for client requests.

3. When a client request is received, assign identifier i to the client and store the identifier in the queue.

4. If multiple client requests are received, then repeat Step 3 for each client.

5. The Server refers the queue and serves corresponding client for a stipulated amount of time and moves to next client after the completion of the predefined time slot.

6. If connection between client and server terminates, then corresponding client identifier present in the queue is deleted and Go to step 5.

7. If queue is not empty then go to Step 2.

8. Stop

APPLICATIONS OF THE SYSTEM

The proposed system can have different applications like –

1. Resource Sharing: Resources are held by the server and thin clients use these resources.
2. Mobile Gaming: Games are installed on server and mobiles can access these games without installing them.

3. Centralized access to client: Clients in an organization can run all the application located on the server irrespective of the client's underling architecture.

CONCLUSION AND FUTURE WORKS

In this paper, we have proposed a novel server based desktop virtualization system that supports mobile devices as thin clients and desktop PC as Server. This system will help clients to execute applications that otherwise would not execute at their devices. This system will support multiple clients and allocate resources using intelligent scheduling algorithms. Further work includes complete implementation of the proposed system with improved algorithms for scheduling at the server-side. More work is also needed to speed up the communication between clients and the server.

ACKNOWLEDGEMENT
We are thankful to our mentor Prof. Sangita Chaudhari for her valuable guidance.

REFERENCES

[1] Citrix Corporation, "Citrix Application Delivery Infrastructure."

[2] Desktop virtualization and security: a global market research report by Citrix.

[3] Desktop Virtualization and,Independent Computing By Dan Kusnetzky, Principal Analyst

[4] XenApp and XenDesktop: Application Virtualization and Desktop Virtualization are Better Together

[5] Intel IT: Embracing Desktop Virtualization *Balancing Employee*

[6] Microsoft Corporation, .Windows Terminal Services.

[7] "3 Key Phases to Implementing Desktop Virtualization" A White Paper Prepared for Citrix

[8] Microsoft Corporation, .Remote.

[9] IEEE PAPER "A Service Based Lightweight Desktop Virtualization System" by Li Yan.

[10] "Desktop Virtualization and Evolving Strategies for IT Service Delivery" by Jaime Halscott