An Efficient Application Virtualization Mechanism using Separated Software Execution System

Su-Min Jang, Won-Hyuk Choi and Won-Young Kim

Cloud Computing Research Department, Electronics and Telecommunications Research Institute, Daejeon, Korea
jsm@etri.re.kr, whchoi@etri.re.kr, wykim@etri.re.kr

Abstract

With the development of cloud computing and virtualization technologies, application virtualization becomes an innovative software usage model. However, the existing solutions show a very limited performance on 3D graphics applications. To solve the problems, we propose a new application virtualization mechanism using separated software execution system. The proposed system does not execute 3D rendering commands or graphic commands of the 3D graphics application at the hosted server but delivers them to client via the Internet and executes them using the client’s graphic device. This mechanism enhances the hosted server’s performance for desktop virtualization service of 3D rendering applications. In performance experiment, the proposed method showed good scalability and obtained a more stable frame rate than the existing desktop virtualization solutions.

Keywords: Application virtualization, cloud computing, 3D graphic application

1. Introduction

In recent, many IT users won’t have powerful machines but they are interested in powerful IT services. The answer to this demand lies with application virtualization using cloud computing and virtualization technologies [1, 2, 3]. It becomes an innovative software usage model that has many benefits as follows.

- Cost reduction
- Better hardware utilization
- Centralized management
- Minimizing outage during maintenance
- Faster deployment of new applications.

Two well-known solutions for the application virtualization are Citrix XenApp [4] and Microsoft RemoteApp [5]. They have the virtualization interface at the hardware abstraction layer. They virtualize common PC hardware like processor, memory and peripheral I/O devices such that multiple operating system instances of different type can be installed on a single machine. However, they are not efficient as 3D rendering service for multiple user service. The reason is that server has a high overhead on 3D rendering service for multiple users, because the server executes all related data processing commands and graphic commands for 2D and 3D and additionally sends the screens data of their results. To solve this problem, we propose a new system denoted as separated software execution system.
(SSES) for software as a service. The proposed SSES can stably support multiple users at the same time, through executing graphic commands at client part.

This paper is organized as follows. Section 2 presents the related work. Section 3 proposes separated software execution system (SSES). Section 4 contains various experimental evaluations. Section 5 concludes the paper with directions for future works.

2. Related Works

2.1 Microsoft RemoteApp

Microsoft Virtualization Desktop Infrastructure accelerates and extends desktop and application deployments to any device. It improves remote worker efficiency, while helping keep critical intellectual property secure and simplifying compliance with regulations. With Windows Server 2012 Remote Desktop Services, VDI (Virtual Desktop Infrastructure) provides a single infrastructure to enable both virtual and session-based desktops and RemoteApp programs.

Figure 1 shows the remote desktop services of Microsoft RemoteApp. Client PC can use virtual applications of host server with RDP (Remote Desktop Protocol). Remote Desktop Services provides functionality similar to a terminal-based, centralized host, or mainframe, environment in which multiple terminals connect to a host computer. In the Remote Desktop Services environment, an application runs entirely on the Remote Desktop Session Host server. The server transmits the graphical user interface to the client. The client transmits the user's input back to the server.

![Figure 1. Microsoft Remote Desktop Services with RemoteApp](image)

RemoteFX vGPU of Microsoft RemoteApp allows multiple virtual desktops to share a single GPU on a Hyper-V server. This GPU sharing leads to a lower performance, as the number of users increases.
2.2 Citrix XenApp

Citrix XenApp is an on-demand application delivery solution that manages applications in the data center and delivers them instantly to users anywhere using any device. XenApp enables IT to centrally manage a single instance of each application and deliver it to users for online and offline use, providing a high definition experience. All service applications are installed and are run on host server. The host server sends screens of application to various client terminals such as phone, desktop PC, notebook, PDA and mouse clicks and keystrokes occurred from client terminal are delivered to the server. The merit of Citrix XenApp is that applications can be accessed from most any device and covering both wired network and wireless network. In particular, the HDX (High Definition User Experience) of Citrix enables to optimize the performance of graphics-intensive 2D and 3D and media rich applications.

Citrix XenApp is extremely limited in providing multiple users with a virtual application service including 3D rendering applications, because Citrix XenApp exclusively uses one GPU of the server for processing 3D rendering commands for one user.

3. Separated Software Execution System

To be seen 3D rendered screens in low performance client, above-mentioned commercial services, Citrix XenApp and Microsoft RemoteApp compress and send result screens of host server. In other words, they use exclusively the GPU of host server for processing 3D rendering commands. Such mechanism is hard to provide many users to graphics-intensive 2D and 3D and media rich applications [6, 7, 8]. To solve the problems, we propose a new application virtualization mechanism using separated software execution system (SSES).

Figure 2 shows the separated software execution system structure. The difference of the proposed SSES mechanism and previous related services is that it is not execute 3D graphic commands of application at the hosted server but delivers them to client via the Internet and executes them using the client’s graphic device. Application virtualization agent of SSES consists of rendering command hooker, virtual object manager and user storage manager. To overcome the overload of host server, the core of our mechanism is the GPU offloading.

![Figure 2. The Proposed System Structure](image)
3.1 Rendering Command Hooker

Figure 3 shows virtual application workflow of SESS. Graphical commands are virtually executed at the host server but are rather delivered to the client through the network by virtual application host modules, and they are really executed at client by virtual application client modules. It means that server don’t make a display of the application.

![Figure 3. Virtual Application Workflow of Separated Software Execution System](image)

When CreateWindowEx() graphical command is called from the application, the processing sequence of the CreateWindowEx() command in SSES is following steps.

1. CreateWindowEx() API generated from the graphics application of host server is intercepted by the rendering command hooker.

2. The virtual application host modules delivers the intercepted rendering command (CreateWindowEx()) to the client through the network.

3. The client creates a window with the execution of the delivered rendering command (CreateWindowEx()) and manages a real handle or real interface instance generated from the API executions.

4. The client sends the return values of the API executions and input data such as those from their mouse and keyboard to the server.

5. The server applies the return values to the host application and maintains a virtual handle or virtual interface instance generated from the fake execution of CreateWindowEx().

The hooking mechanism of separated software execution system is a type of using wrapper libraries. When executing application of host server calls graphic commands, our hooking module of wrapper library are executed instead of original library. When creating a wrapper, we make our own version of a library that an application loads, with all the same functionally of the original library that it will replace. When executed application on SSES server calls graphic commands such as CreateWindowEx(), Direct3DCreate9 and D3D10CreateDevice, our SSES server hooking module executes our wrapper library instead of original library.
3.2 Virtual Object Manager

The proposed system needs two separated working processes. One process of the client is executed for rendering commands, and another is executed for non-graphic tasks at the server. The two processes should work closely, almost as one process. We should maintain paired objects that are individually generated from the client and the server. Figure 4 shows paired objects of host server and client PC. These objects include windows handles, file pointers, string data and etc. They are generated from working processes.

![Figure 4. Virtual Object Manager of Seperated Software Execution System](image)

To quickly access the paired object, we use a stack (LIFO) and tree-index. Frequently referenced objects are accessed by the stack, and the remaining objects are accessed by the tree-index.

3.3 User Storage Manager

Figure 5 shows the user storage manager of separated software execution system. To support for multiple users, we separate logically SSES into three servers (user info and SW management server, storage server and SW executing server). The access of user storage data is separated with two file filter system. These are the user storage access of normal application and that of SSES application. For security access of user storage data, we block normal application’s access to user storage data.

![Figure 5. User Storage Manager of Seperated Software Execution System](image)
The storage server is separated from SW executing server in order to minimize duplication of user specific data. The user storage manager consists of remote storage handler module, storage security controller module, network drive handler module, and storage server resource monitoring module.

4. Performance Evaluation

We compared SSES with Citrix XenApp and Microsoft RemoteApp for performance evaluation of the proposed method. Used test programs are Archispace LT and MultiAnimation program as shown in Fig 6. MultiAnimation sample program has APIs that handle the loading of the mesh to be animated, as well as the blending of multiple animations. ArchiSpace LT is a commercial application used for generating and displaying virtual buildings or architectural drawings using 3D objects.

![MultiAnimation](a) MultiAnimation ![ArchiSpace LT](b) ArchiSpace LT

Figure 6. Screenshots of the Test Programs

The clients and servers are connected with a 100 Mbps Ethernet. Table 1 shows the specifications of the clients and servers.

<table>
<thead>
<tr>
<th></th>
<th>CPU / Memory</th>
<th>GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>Intel Core i7 3.4GHz, Memory 8G</td>
<td>NVIDIA GeForce GTX 570</td>
</tr>
<tr>
<td>Citrix server</td>
<td>Intel Core i7 3.4GHz, Memory 16G</td>
<td>NVIDIA Quadro FX3800 4GB * 4 ($5000)</td>
</tr>
<tr>
<td>MS server</td>
<td>Intel Xeon X5650 2.6GHz Memory 20G</td>
<td>NVIDIA Quadro 6000 8GB ($5500)</td>
</tr>
<tr>
<td>SSES server</td>
<td>Intel Core i7 3.4GHz, Memory 16G</td>
<td>NVIDIA GeForce GTX 480 ($450)</td>
</tr>
</tbody>
</table>

Table 1. The Specification of the Clients and Servers

We evaluate frames per second (FPS) and GPU usage rate on each server (MS, Citrix, SSES server), where the program used is MultiAnimation and single client connects to the each server. SSES shows a more stable FPS than MS and Citrix as shown in Fig. 7(a). On the other hand, MS shows a large gap between the maximum and minimum FPS, which becomes the cause of an unstable display. Figure 7(b) shows that the GPU usages of MS and Citrix are very high, but the SSES rarely uses them since the rendering commands are executed not at the server but at the client.
Figure 7. Evaluation of Single Client

Figure 8 shows an evaluation of multiple clients on MS server and SSES server. Ten clients connect to the server at the same time, where the program used is Archispace LT. The evaluated factors are frames per second (FPS). When the number of clients is 10, all clients on MS server show the same FPS, which decreases significantly, as MS server executes all of the rendering and non-graphic tasks. However, the FPS of a SSES client is independent of the client count, since each client uses its own GPU.

Figure 8. Evaluation of Multiple Clients

5. Conclusion

In recent, application virtualization becomes an innovative software usage model. Therefore, application virtualization has been widely used in the multi-user service. However, the existing solutions show a very limited performance on 3D graphics applications. To improve the capability of the application virtualization services, we have devised the smart separated software execution system (SSES) which is more effective on service for multiple users. In performance experiment, the proposed SSES shows a better scalability and more stable frame rate than existing virtual desktop solutions under various applications. In the near future, the proposed mechanism will be applied to various real applications and will be evaluated in various environments.
Acknowledgements

This work was supported by the Technology Innovation Program [10035185, Development of a Separated Software Execution Technology for Software Service] funded by the Ministry of Knowledge Economy (MKE, Korea).

References


Authors

Su-Min Jang

He received his B.S. in Computer Science and Statistics from Mokpo National University, Korea in 1997 and he received M.S. and Ph.D. in Information and Communication Engineering from Chungbuk National University, Korea in 1999 and 2007, respectively. He works now at the ETRI, Korea since 2011. His main research interests include distributed virtual environments system, location based services, game server.

Won-Hyuk Choi

He received his B.S. in Computer Engineering from Kyungpook National University, Korea in 1999 and he received M.S. in Computer Engineering from Kyungpook National University, Korea in 2001. He works now at the ETRI, Korea since 2001. His main research interests include software virtualization, software service and graphic offloading system.

Won-Young Kim

She received his B.S. in Computer Science from Ewha Womans University, Korea in 1989 and she received M.S. and Ph.D. in Computer Science from Korea Advanced Institute of Science and Technology, Korea in 1991 and 1998, respectively. She works now at the ETRI, Korea since 1999. Her main research interests include software service.