

Method for Distribution, Execution and Management of the Customized Application based on Software Virtualization

Kyong-I Ku, Won-Hyuk Choi, Moonyoung Chung, Kiheon Kim, Won-Young Kim, Sung-Jin Hur

SW Service Research Team, Electronics and Telecommunications Research Institute, Korea

{kiku, whchoi, mchung, khkim50, wykim, sjheo}@etri.re.kr

Abstract— Software virtualization allows applications and data to be put into virtual layers. We propose the method for distributing, executing and managing the customized application based on software virtualization. Customized application is composed of one or more applications, add-ons, plug-in and other various settings associated with software virtualization technique. Customized application generates the virtual boundary which separates and isolates the hard disk, memory and other resources at the execution time with a well-informed area to allow the user to write some data. So, our method ensures that customized application use a personal computer without modifying the hardware resource and without interfering with other applications in a safe state.

Keywords— Customized Application, Software Virtualization, Optimized and Secured application

I. INTRODUCTION

To perform any work in personal computer, several applications are needed. At that time, any applications require very specific versions of other applications. And there is a need of extensions throughout application's add-ons and plug-ins. User may set the configuration of application to satisfy user preference. Moreover, some applications do not work very well under current status of user's personal computer without cause. Frequent changes of application environment like installation and uninstallation makes an application unstable. Moreover, an online automated software deployment and update may threaten the performance, security and reliability of system. For example, malicious codes on web browser are spreading through automatic online update. Use of unsecured application at the enterprise where needs the integrity at work is a sensitive issue.

Software virtualization allows applications and data to be put into virtual layers (application layers and data layers) instead of being installed to the base file system and registry ([1]–[6]). To do that, software virtualization isolates the application from the various resources such as files, registry, and so on. At execution time, software virtualization installs an application at the virtual layer and mimics the normal behaviour of one. In this paper, we propose the method for

distributing, executing and managing the customized application based on software virtualization. Customized application is composed of one or more applications, add-ons, plug-in and other various settings of it (validation, write restrictions, and so on) associated with software virtualization technique. The distributor creates a customized application using customized application converter and distributes it. User deploys a customized application and executes it. Customized application generates the virtual boundary which separates and isolates the hard disk, memory and other resources at the execution time. Customized application runs at a personal computer without modifying the hardware resource and without interfering with other applications. Moreover, customized application has a well-informed area to allow the user to write some data at a specific folder and to restore the initial state of customized application. So, it ensures the stable and flexible computing environments.

The rest of this paper is organized as follows. In Section 2, we give a brief survey on technologies related to the subject. In Section 3, we summarize the customized application. And we propose the method for distribution, execution and management of the customized application based on software virtualization. Finally, we conclude this paper and present future works in section 4.

II. RELATED WORKS

Application virtualization is an umbrella term that describes software technologies that improve portability, manageability and compatibility of applications by encapsulating them from the underlying operating system on which they are executed [7]. It allows applications to run in environments that do not suit the native application. It also improves security, by isolating applications from the operating system.

Software virtualization is also known as application virtualization. It allows applications and data to be put into virtual layers instead of being installed to the base file system and registry[1]. To do that, software virtualization isolates the application from the various resources such as files, registry, and so on. At execution time, software virtualization installs an application at the virtual layer and mimics the normal behaviour of one as shown in Figure 1.

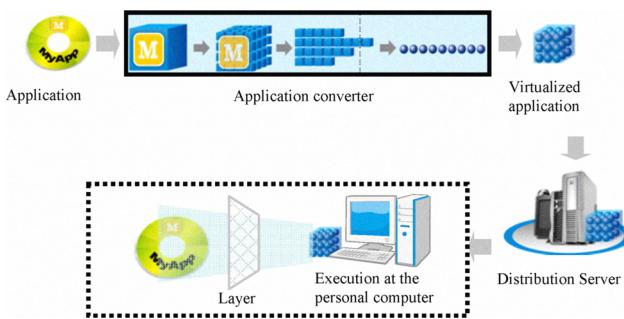


Figure 1. Behavior of software virtualization

There are some vendor-specific implementations. VMware ThinApp[2], Software Virtualization Solution[3], AppStream[4], PortableApps[5], AutoIt[6] and Microsoft Application Virtualization[8] are examples of application virtualization.

VMware Thinapp[2] known as Thinstall is a solution that generates a single execution file that is isolated from the OS. When a user runs this single execution file, it performs file and registry I/O by using application level virtualization. Software Virtualization Solution[3] allows the user to install an application without modifying the configuration. To do that, Software Virtualization Solution captures all changes made during installation of a particular application in what's called a Virtual Software Package.

Symantec Workspace Streaming[4] as known as AppStream[4] adds a streaming function to the software virtualization function. It enables software to be executed during on-line transmission time.

PortableApps[5] provides packaged software suitable for use on portable media, without leaving settings files on the host computer using NSIS script. NSIS script is a script-driven Windows installation system with minimal overhead backed by Nullsoft.

AutoIt[6] is a freeware BASIC-like scripting language designed for automating the Windows GUI and general scripting. It uses a combination of simulated keystrokes, mouse movement and window/control manipulation in order to automate tasks in a way.

Microsoft Application Virtualization[8] known as App-V enables organizations to respond to the management challenge by providing the capability to make applications available to end-user computers without having to install the applications directly on those computers.

Software management is one of the most time-consuming aspects of managing an enterprise network environment. As described above, there are various promising software virtualization technologies. Software virtualization technologies offer several new technologies to reduce the complexity of managing applications in enterprise environments. However, virtualized application has no rights to update itself manually, the user may have restrictions on the use of application.

III. CUSTOMIZED APPLICATION BASED ON SOFTWARE VIRTUALIZATION

Software virtualization allows applications and data to be put into virtual layers. We propose the method for distributing, executing and managing the customized application based on software virtualization to ensure the stable and robust computing environments.

A. Customized Application

Customized application is composed of one or more applications, add-ons, plug-in and other various settings (validation, write restrictions, and so on) associated with software virtualization technique.

Web browser application is a software application for retrieving, presenting, and traversing information resources on the World Wide Web. Web browser application has a variety of plug-ins and many extensions. If the system manager customizes the web browser with mandatory plug-ins and distributes it as a customized application, the member can work under the secure environment without the effect on the personal computer. Developer installs and configures two or more complex applications to build the development environments. If the company distributes these applications together as a customized application, other developers can easily deploy the development environments and conveniently use it. The example of customized application is shown in Figure 2.

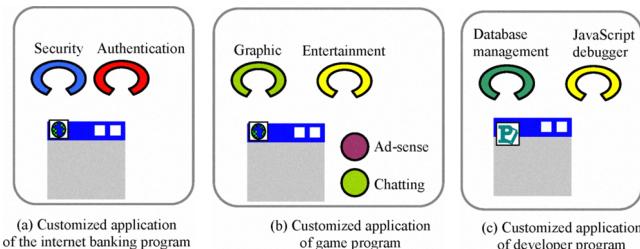


Figure 2. Example of the customized application

At the bank, the system manager wants to distribute the customized internet banking program. The system manager includes the security module and authentication module in web browser. And the system manager of a bank adds several configuration values and restricted workspace of users. The system manager of a bank makes the customized application (including web browser) and distributes it to bank customers as shown in Figure 2(a). A game company can also distribute a customized application with graphic module, entertainment module, ad-sense plug-in and chatting plug-in as shown in Figure 2(b). A software development company can also distribute a customized application to satisfy other developers a Figure 2(c).

Customized application has a well-informed area to allow the user to write some data at a specific folder and to restore the initial state of customized application. It allows the user to update the application, if user wants. And it allows the user to reload the application of initial state, if user wants. This was

very helpful that the user wants to add personal plug-ins or add-ons to customized application.

Customized application generates the virtual boundary which separates and isolates the hard disk, memory and other resources at the execution time. Therefore, customized application based on software virtualization technologies has the following characteristics:

First, the customized application is running on user's personal computer without installation. Installation and uninstallation of customized application don't affect the user's personal computer. It acts as 'file copy'. In other words, installation, execution and uninstallation of customized application don't change the state of the user's personal computer.

Second, the customized application can be composed of the company's software applications selected by the administrator, one or more additional add-ons and plug-ins, and other necessary settings. By doing so, system manager of the company can distribute the optimized and secured application to the members. The member doesn't need additional work on application. The company can control the members' personal computer effectively.

Third, as the customized application has the restriction on write operation, the change of application can be limited. By doing so, regardless of the user's intent, system manager controls the extension of the add-ons or plug-in on customized application. But it also supports the update of application, if user wants.

Fourth, if necessary, the customized applications can be restored to its initial and secure state. In this case, change of the application and add-on or plug-in added by the user are ignored.

B. Architecture of Customized Application

Architecture of customized application is shown in Figure 3. Customized application converter is used to generate customized application as described in [11]. [11] is the example of generation of customized application that is composed of only one application. Customized application is a single execution file that is made of *application execution logic* and *application repository*. Customized application generates the virtual boundary of application to separate and isolate the hardware resources at the execution time.

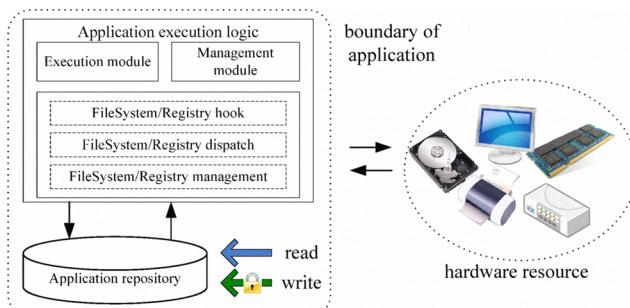


Figure 3. Architecture of customized application

Application execution logic is the unique format of the file which is described in [11]. It includes the files which run applications, registry and shared components. *Execution module* on application execution logic runs customized application. *Management module* on application execution logic manages customized application. *FileSystem/Registry hook* on application execution logic hooks library calls at user-level. *FileSystem/Registry dispatch* on application execution logic dispatches library calls to appropriate destination(virtual registry, virtual file system, real registry, real file system, or write-enable application repository). *FileSystem/Registry management* on application execution logic manages file system and registry.

Application repository, the storage of applications' image file, manages the file as the unit of blocks. It supports the write operation of blocks. Application repository assigns the blocks to save a file and releases the blocks to delete a file. Also, it includes the shared information of files, registry and common components. This information has the rights of read-write operation or partial rights of write operation. So, the customized application can support the extension of applications, change of configuration, creation of data through write operation. Moreover, if necessary, the customized application can be restored at the initial state.

C. Execution of Customized Application

As described above, several applications are encapsulated in a single execution file to compose a customized application. User clicks the customized application to do work. Execution module(daemon) on customized application is extracted and stored in any folder. It runs. Execution module extracts the actual running applications from the application repository and runs it associated with itself. *FileSystem/Repository hook*, *FileSystem/Registry dispatch* and *FileSystem/Registry management* binds to execution module. Customized application encapsulated in a single execution file is executed as shown in Figure 4.

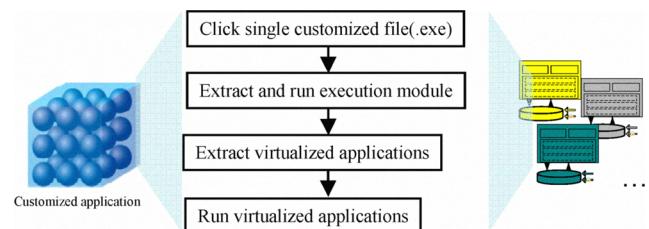


Figure 4. Execution of customized application

Customized application generates the virtual boundary which separates and isolates the hard disk, memory and other resources at the execution time. We explain the execution path of the customized application. General Windows's applications accessed the system hardware through system libraries and the OS kernel as shown in Figure 5(a)[9,10]. It will affect the system performance, security and stability. In contrast, customized application behaves as shown in Figure 5(b). *System library filter* of the customized application inspects the

pre-conditions before the calls of system library and inspects the post-conditions after the calls of system library.

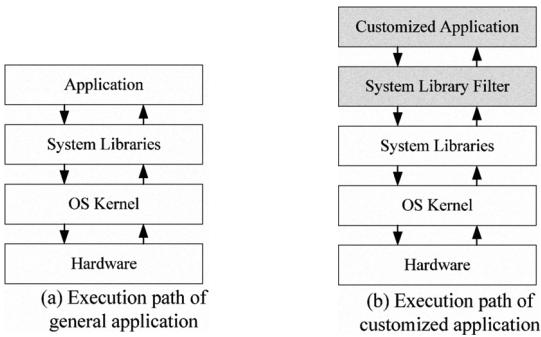


Figure 5. Execution path of general application and that of customized application

System library filter acts as shown in Figure 6. System library filter intercepts the calls of system library and inspects the function parameters and other data to examine whether the actual function can be called.

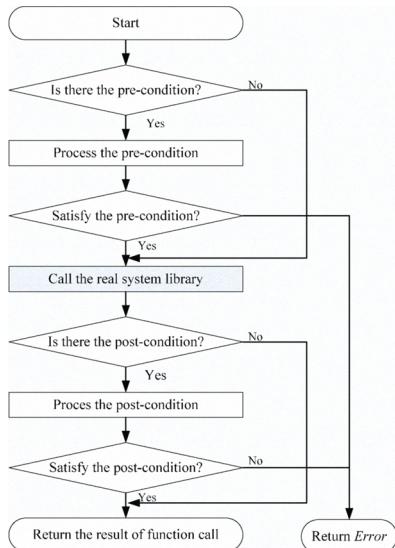


Figure 6. System library call of system library filter at customized application

Pre-conditions are reviewed before the function call. If it is an appropriate function call, the system library filter of customized application calls the real function of system library. Otherwise, it returns the error. For example, the customized application wants to write some data at a specific folder. Pre-condition can be checked. Does it allow a write operation at this folder? If yes, the customized application can create, read and write the file at a particular folder. Also, the write operation of read-only registry can not be allowed to write by the system library filter.

Post-conditions are reviewed after the function call. If it is an appropriate function call, the system library filter of

customized application returns the result of system library. Otherwise, it undoes the function call and returns the error.

IV. CONCLUSIONS

In this paper, we propose the method for distribution, execution and management of the customized application based on software virtualization. Software virtualization allows applications and data to be put into virtual layers instead of being installed to the base file system and registry. There are some vendor-specific implementations of software virtualization. Previous techniques have no rights to update themselves; the user may have restrictions on the use of application. Our proposed method, customized application, has a well-informed area to allow the user to write some data at a specific folder and to restore the initial state of customized application. We define customized application and show the architecture of it. Customized application is a single execution file that is made of application execution logic and application repository. Customized application generates the virtual boundary of application to separate and isolate the hardware resources at the execution time. Unlike general Windows's application, customized application calls the system libraries through system library filter and guarantees the secured call. Proposed method was very helpful because it made applications run even when they couldn't run because of older versions or other software that was denying the software to run easily.

In this future, we will show the detailed customized application converter and executer and implement it to a single execution file. So, we will apply our customized application to the software service on a personalized software service platform [12] or another software distribution channels.

ACKNOWLEDGMENT

This work was supported by the IT R&D program of MKE/IITA. [2007-S-015-01, Development of a Personalized SW Service Platform on Movable Disk Devices]

REFERENCES

- [1] http://en.wikipedia.org/wiki/Software_Virtualization_Solution
- [2] VMware ThinApp, <http://www.thinstall.com/>
- [3] Software Virtualization Solution, <http://www.altiris.com/Products/SoftwareVirtualizationSolution.aspx>
- [4] AppStream, <http://www.appstream.com>
- [5] PortableApps, <http://portableapps.com/>
- [6] AutoIt V3, <http://www.autoitscript.com/>
- [7] http://en.wikipedia.org/wiki/Application_virtualization
- [8] <http://www.microsoft.com/isd>
- [9] Won-Hyuk Choi, Won-Young Kim, Ji-Hoon Choi, Wan Choi, "A Design and Implementation of Application virtualization method using virtual supporting system and Copy-on-Write Scheme", Conference of Korean Contents Association, 2007.
- [10] Won Hyuk Choi, Won-Young Kim, Wan Choi , " Design and Implementation of Portable software converting method using Software installation information extraction tool and AutoIt Script", Conference of Korean Society for Internet Information , 2008.
- [11] Won Hyuk Choi, Won-Young Kim, Sung-Jin Hur , "A Method of SW Installation and Execution Information Extraction using Virtual Execution Handler and Multiple Virtual Execution Environments on Dirty PC Environment," 1st International Conference on Internet (ICONI) 2009.
- [12] Personalized SW service platform Youfree, <http://www.youfree.or.kr>