



The Study: Disk Information Acquisition in QEMU-based Virtualization

Jihoon Moon¹, Hoichan Jeong²

^{1,2}eReD Team of Security Labs, SOOSAN INT

Abstract—This paper explains how to obtain disk and volume information by using a virtual disk in the QEMU/KVM environment of the Linux operating system. In some cases, one may require disk and volume information for a specific virtual disk. To enable this, we designed a function for obtaining disk volume information by using the libguestfs API, and reported disk volume information results for a specific virtual disk file.

Keywords—component; QEMU, KVM, Hypervisor, Libguestfs, Qcow2

I. INTRODUCTION

Virtualization manages multiple virtual machines using a hypervisor. In some cases, we need to manage virtual machines from the outside. To enable this, we examined how a virtual disk file can be mounted on the host. We confirmed that disk and volume information can be acquired by using the libguestfs library in the virtual disk file used in the virtual machine.

The remainder of this paper is organized follows. Chapter 2 explains how to mount virtualized disk files on the host. Chapter 3 introduces the function for obtaining virtual disk and volume information through a virtual disk file. In Chapter 4, experiments on obtaining disk information are reported. Finally, Chapter 5 concludes the paper.

II. RELATED WORK

2.1. FUSE

FUSE (FileSystem in Userspace) ^[1] is an application program that uses a kernel module (FUSE module) and a library (libfuse), and it plays a role in the file system ^[2]. The application sends a file-related system call to the directory where FUSE is mounted. The system call passes through the Linux Virtual File System (VFS) and is delivered to the FUSE module in the Linux kernel. The FUSE module transfers the system call to the corresponding FUSE daemon process using the target directory of the received system call and the FUSE mount information. The FUSE daemon process processes the corresponding system call and sends the result to the application process. Because the application generates two IPCs each time a system call is made, FUSE is generally slower than a normal file system. The virtual disk image can be mounted using the qcow2-fuse library ^[3].

2.2. libguestfs

libguestfs is a project developed by Red Hat Inc. This library can access and modify virtual machine images. It can access all known types of Linux, Windows, Mac OS X, BSD file system and LVM2 volume management, MBR and GPT disk partitions, qcow2, VirtualBox VDI, VMD-VHDX, VMD, and VHD ^[4].

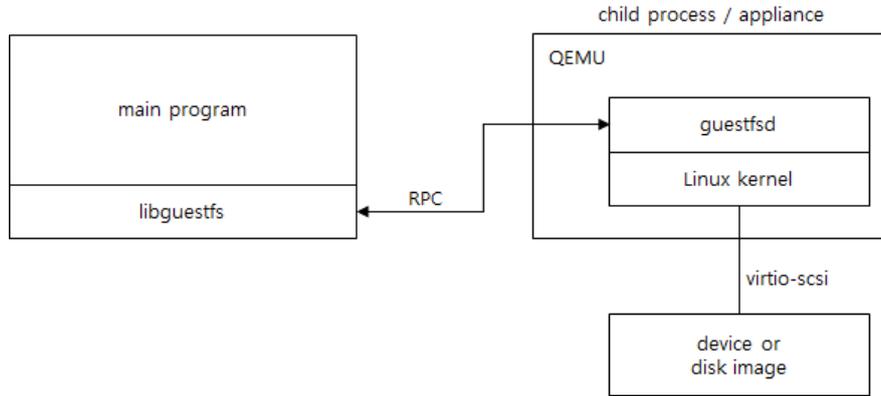


Figure 1. libguestfs architecture

Figure 1 illustrates the system architecture of libguestfs. QEMU runs as a child process of the main program. The inner module consists of the Linux kernel, stack, and guestfsd. The libguestfs library enables many of the functions that QEMU can do without modifying QEMU. However, this requires many resources because a new QEMU instance must be started every time a virtual disk is opened. libguestfs includes built-in scripting tools to modify and view the virtual disk image internally. It also recognizes the disk-related virt-df [5] and geustfish [6-7], which enable control over files on the disk.

```

ered@ered-OptiPlex-7040:~$ virsh list --all
Id      이름                상태
-----
-       winserver2016        끝 내 기

ered@ered-OptiPlex-7040:~$ virt-df -d winserver2016
Filesystem            1K-blocks      Used    Available    Use%
winserver2016:/dev/sda1    511996        367476    144520        72%
winserver2016:/dev/sda2  20457468     13060476  7396992        64%
    
```

Figure 2. check disk utilization using the virsh-df command

```

ered@ered-OptiPlex-7040:/mnt/vms$ guestfish -a win10pro_19052301 -i

Welcome to guestfish, the guest filesystem shell for
editing virtual machine filesystems and disk images.

Type: 'help' for help on commands
      'man' to read the manual
      'quit' to quit the shell

Operating system: Windows 10 Enterprise
/dev/sda2 mounted on /

><fs> ls /
$Recycle.Bin
Documents and Settings
PerfLogs
Program Files
Program Files (x86)
ProgramData
Recovery
System Volume Information
Users
Windows
pagefile.sys
swapfile.sys
><fs>
    
```

Figure 3. check image directory using guestfish command

The virt-df command is one of the utilities created when the 'libguestfs-tools' package is installed. This command can be used to check the file system type and disk capacity information of the disk

partition. An example of its use is shown in Figure 2. Figure 3 shows the screen for checking the virtual disk directory using the `guestfish` [6-7] command.

2.3 QEMU-NBD

The QEMU-NBD function has been added to QEMU version 0.10. It is a system that can remotely access block devices as if they were local. Figure 4 shows the structure of QEMU-BSD. The server system to which the block device is connected must run the NBD-SERVER program to make it available to the client. The client system then runs `nbd-client` to create a new block device (`/dev/nbd0`) [8].

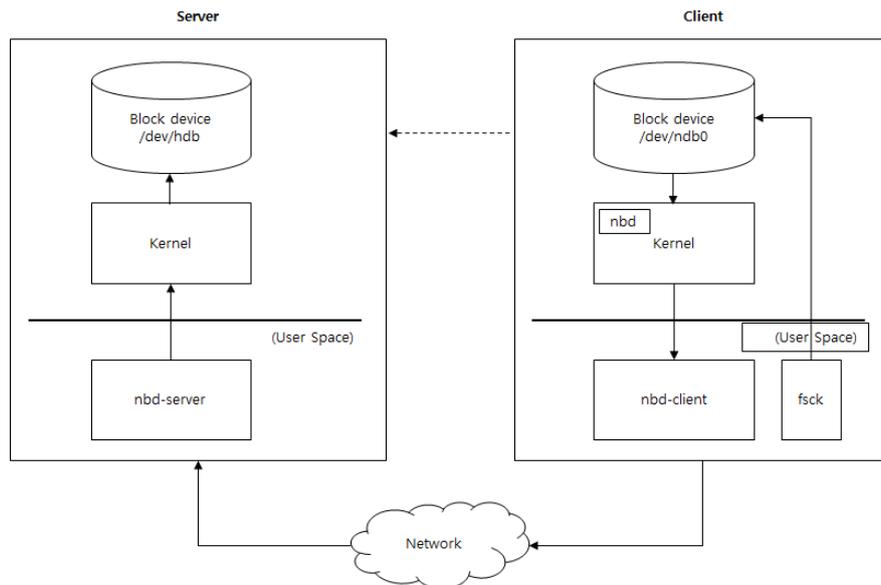


Figure 4. QEMU-BSD architecture

2.4 iSCSI

The internet small computer system interface (iSCSI) protocol is an Internet Engineering Task Force standard protocol that encapsulates SCSI commands into TCP / IP packets to support block data transmission over IP networks [9].

III. PROPOSED ARCHITECTURE

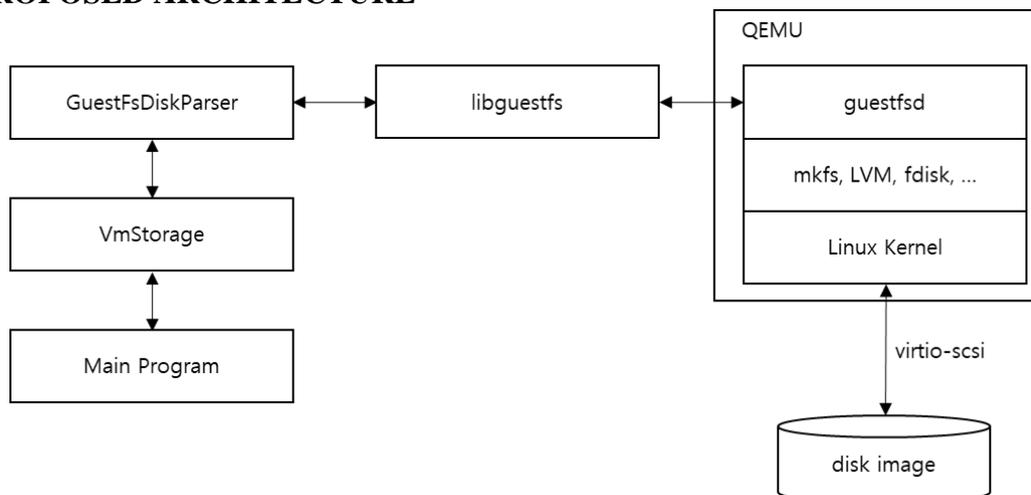


Figure 5. System Architecture

Figure 5 illustrates the system structure for obtaining disk and volume information from the proposed virtual disk image. Virtual disk and volume information can be acquired using a disk image file in a virtual machine. In this figure, VmStorage is a module that performs disk functions and recovers disk and volume information through GuestFsDiskParser. GuestFsDiskParser is a module that works with the libguestfs library.

IV. EXPERIMENTAL RESULT

In the virtual disk image used in the virtual machine, the disk information and internal information of the volume were tested by linking with the libguestfs API. Table 1 describes the experimental setup.

Table 1. Test Specification

Hardware	Host Server	4 core CPU 16GB Memory Samsung 500GB SSD
	Virtual Machine	2 Core vCPU 2GB Memory 20GB Virtual HDD
Software	Operating System	Host : Ubuntu 16.04 VM : Cent OS 7.0
	Hypervisor	QEMU/KVM 2.5.0

Figure 6 shows the test results for the experiment. (1) represents the device and volume information used in the virtual disk image. (2) represents the partition type and partition identifier information for the MBR. (3) can obtain a volume ID, operating system type, and other information with respect to the number of partitions possessed by one virtual disk image. The operating system type is 0 for unknown, 1 for Windows, and 2 for Linux.

```

[guestFs::DiskContainer::Refresh@72] InitDomain()
[guestFs::DiskContainer::Refresh@80] m_diskCount=1
[guestFs::DiskContainer::Refresh@85] loop_index=0
[guestFs::VmDisk::VmDisk@18] m_path=/dev/sda, m_index=0
[guestFs::GuestFsApi::ListFilesystem@193] path=/dev/sda1, type=xfs (1)
[guestFs::GuestFsApi::ListFilesystem@193] path=/dev/centos/root, type=xfs
[guestFs::GuestFsApi::ListFilesystem@193] path=/dev/centos/swap, type=swap
[guestFs::VmDisk::VmDisk@18] m_path=/dev/sda, m_index=0
[protSvc::VmDiskParser::VmStorage::RefreshVolumes@140] Volume Refresh Start
[guestFs::GuestFsApi::ListFilesystem@193] path=/dev/sda1, type=xfs
[guestFs::GuestFsApi::ListFilesystem@193] path=/dev/centos/root, type=xfs
[guestFs::GuestFsApi::ListFilesystem@193] path=/dev/centos/swap, type=swap
[FilterMountableFilesystems@186] File System List Cnt:
[GetMbrPartType@114] device=/dev/sda, partNum=1
[GetMbrPartType@118] mbrPartType=primary (2)
[guestFs::GuestFsApi::PartGetMbrId@642] MBR type byte=131
[FilterMountableFilesystems@186] File System List Cnt:
[FilterMountableFilesystems@186] File System List Cnt:
[guestFs::DiskParser::InitVolumes@398] diskIndex=0
[guestFs::DiskParser::Init@345] index=0, Size=16106127360
[protSvc::VmDiskParser::VmStorage::RefreshVolumes@147] volumeCount=2
[protSvc::VmDiskParser::VmStorage::RefreshVolumes@151] =====
[protSvc::VmDiskParser::VmStorage::RefreshVolumes@154] Partition Num=0
[protSvc::VmDiskParser::VmStorage::RefreshVolumes@163] volumeId=B12C6A0856B14C5C9B4690632B25E787 (3)
[protSvc::VmDiskParser::VmStorage::RefreshVolumes@167] OS Type=2
[protSvc::VmDiskParser::VmStorage::RefreshVolumes@154] Partition Num=1
[protSvc::VmDiskParser::VmStorage::RefreshVolumes@163] volumeId=8855368169DC494A9A444D1625CC72A7
[protSvc::VmDiskParser::VmStorage::RefreshVolumes@167] OS Type=0
[protSvc::VmDiskParser::VmStorage::RefreshVolumes@169] =====
[protSvc::VmDiskParser::VmStorage::RefreshVolumes@170] Volume Refresh End
    
```

Figure 6. Test Result

V. CONCLUSION

In this study, we investigated how to obtain disk and volume information from a virtual disk image in a Linux KVM-based virtualization environment. The results of this study showed that among the methods of acquiring virtual disk information, the method of using the libguestfs library is capable of obtaining much more information about disks and partitions.

REFERENCES

- I. FUSE: Filesystem in Userspace, [Online]. Available: <http://fuse.sourceforge.net>
- II. Tae-Yeong Son, Seong-Rak Rim, FUSE-based Syslog Agent for File Access Log, JKAIS Vol.17 no.7, pp. 623-628, 2016.
- III. libfuse, [Online]. Available: <https://github.com/libfuse/libfuse>
- IV. libguestfs, [Online]. Available: <https://en.wikipedia.org/wiki/Libguestfs>
- V. virt-df, [Online]. Available: <http://libguestfs.org/virt-df.1.html>
- VI. guestfish, [Online]. Available: <http://libguestfs.org/guestfish.1.html>
- VII. Richard W.M. Jones, libguestfs tools for viewing and modifying virtual machine disk images, KVM Forum, Nov 2012.
- VIII. Francesc Zacarias Ribot, QLOOP Linux driver to mount QCOW2 virtual disks, M.S Thesis, Barcelona School of Informatics University, Jun 2010.
- IX. U.-R. Jeong, Performance Analysis of iSCSI SAN System including Host-Caching, M.S Thesis, Hannam University, Feb 2014.