

Virtual Data Storage Management based on KVM and Docker

Huibert Kwakernaak¹ and Pradeep Misra²

Abstract—This electronic document is a live template. The various components of your paper [title, text, heads, etc.] are already defined on the style sheet, as illustrated by the portions given in this document.

I. INTRODUCTION

In the process of developing products, the code is usually tracked by using Git, use the continuous integration(CI) and continuous delivery(CD) to implement project automated construction, testing and deployment cycles to rapidly interact with high-quality products. Virtual machine is used to deploy applications, some domestic companies can provide virtual server services, such as Alicloud, Qcloud etc. Many companies use virtual servers to deploy their own applications, which can save costs and increase resource utilization. There are also several issues while using virtual servers. The first problem is that the tenancy deploys the application on the virtual machine. If the virtual machine fails, the files in the virtual machine can not be removed and can only be discarded. Discarding data in the big data era is a tremendous loss to the tenancy. The second problem is the amount of data. If use a machine learning method to Train a model, then the amount of data in the learning process is continuously accumulated. However, the capacity of a virtual machine is fixed, there is no way to meet the growing demand for storage of data. Therefore, how to implement a storage system that can be dynamically expanded and separated from the virtual machine is a problem that needs to be solved. The third problem is the sharing of data with authority. For example, When using the same dataset for different studies or using another dataset for further research, the copy method is not an efficient sharing strategy if the amount of data is large. And when sharing data, the owner of the data should set the permissions of the data in order to protect the data security. This paper presents two models to solve the above three problems respectively.

II. TECHNICAL INTRODUCTION AND CHOICE

A. Virtualization technology

Virtualization technology can help us to strengthen IT resources into single unit that can be shared over a network and fulfill the shrinking resources and improve resource utilization[1]. The virtual machine is an abstract entity between the system hardware and the user, like a physical

computer, runs an operating system and applications. Similar to the operating system's process management, each running process considers itself the only process running in the system. The virtual machine is running on the host computer by simulating hardware input and output at the software level. The goal of virtualization technology is simulate another operating systems execution environment on existing operating systems. Common virtualization technologies include Xen, kernel-based virtual machine(KVM)and Hyper-V.

Xen is an open source project developed by the Computer Lab at University of Cambridge. Xen inserts a virtualization layer between the system hardware and the virtual machine, translating the system hardware into a logical compute resource pool, and Xen dynamically allocates resources in the resource pool to the operating system or application. One of the basic principles of Xen system design is the separation of schemes and mechanisms. The VMM in a Xen system is in a mixed mode. An authorized privileged VM exists in the Xen system to help the VMM system. The Xen program to make the schemes and the privileged VM to be responsible for the implementation of the mechanisms. A privileged VM licensed by Xen is called Domian0. Design principles that separate program and mechanism improve the flexibility and performance of Xen systems. Xen system core code is rare, this is because Xen runs at a privileged level higher than the operating system level, if there is a bug in the Xen kernel will endanger the safety of the entire system.

KVM is a full virtualization solution for the Linux kernel. The fundamentals KVM developers followed were the same as the Linux kernel: "Don't reinvent the wheel"[2]. KVM development did not create a hypervisor by changing the Linux kernel code. In contrast, KVM virtualization technology uses the Linux kernel as its hypervisor. Support for KVM virtualization has become the default part of the mainstream Linux kernel since 2.6.20. But KVM is just a module in the Linux kernel, and more tools are needed to manage and create a complete KVM virtual machine. QEMU is a powerful virtualization tools that virtualizes different CPU frameworks. KVM system uses the X86 part of QEMU, after development and transformation to form a user space tool that can control KVM kernel module QEMU. Although QEMU tools can create and manage KVM virtual machines, QEMU is inefficient and not easy for users to use. Therefore, RedHat has developed more tools for users such as libvirt, virsh and virt-manage. Tools libvirt provides a convenient, reliable, multi-language supported programming interface for a variety of virtualization tools. Users can connect to a host on a KVM or Xen system using functions provided by libvirt to control different virtual machines. In addition, virsh is

*This work was not supported by any organization

¹H. Kwakernaak is with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands h.kwakernaak at papercept.net

²P. Misra is with the Department of Electrical Engineering, Wright State University, Dayton, OH 45435, USA p.misra at ieee.org

a set of text-based virtual machine management command tools provided by libvirt. Users can use virsh to implement all the features of libvirt. Auxiliary tools virt-manage is a set of virtual machine management graphical interface software developed in the python language, that is implemented using the libvirt API for intuitive operation of virtual machines.

Hyper-V is a virtualization product based hypervisor. The purpose of Hyper-V is to provide a broader range of users with more familiar, cost-effective virtualization infrastructure software. Hyper-V supports full virtualized guests and paravirtualized guests. Paravirtualized guests requires that the virtual machine and the physical host have the same operating system, and full virtualized guests requires that the physical host's CPU support virtualization. Full virtualized can create virtual machines for different operating systems. But Hyper-V requires that the host system must be Windows.

Because this platform uses a Linux system, it does not use Hyper-V. Both KVM and Xen are open source virtualization technologies that have their own strengths and weaknesses. Xen uses a self-developed hypervisor to manage virtual machines and related resources but KVM directly uses the Linux kernel as a hypervisor. Since KVM is a special module in the Linux kernel, after the Linux kernel loads the module, it can turn the Linux kernel into a hypervisor so that KVM can well schedule and manage hardware resources. For the above reasons, the underlying virtual machine technology in this article uses KVM.

B. Container virtualization technology

Container technology is an operating system-level virtualization solution. Container can divide individual operating system managed resources into isolated groups to balance resource requirements. Compared with virtualization technology, container technology do not require instruction-level simulation and real-time compilation. Common container virtualization technologies include Linux Containers(LXC) and Docker. LXC is a Linux system lightweight container technology that includes resource management and security management. LXC uses the cgroup mechanism and namespace mechanism to implement resource management functions and security management in the Linux kernel. LXC can supports all the features and capabilities available in the Linux environment. Docker is an improved version of Linux Container. Docker encapsulates applications and dependencies into a container, implementing a project migration that requires the container to be started in new environment. Docker containers isolate software from its development environment, such as differences between the development environment and staging environment, which can reduce conflicts between running different software on the same infrastructure.

The main difference between container technology and virtualization technology is in their architectural approach. Virtualization technology is implemented at the hardware level and requires a hypervisor and virtual machine operating system that must load a complete operating system and virtualize system resources at startup. In contrast to virtualization,

container technology is implemented at the operating system level and shares the operating system kernel with the host computer. It does not need to start a complete operating system at startup, so the container boots much faster than the virtual machine. Virtualization technology provides an operating system environment, that is the same as the actual production environment, mainly used to build applications or environments. Traditional virtualization technology usually uses block storage to implement virtual machines data storage, but can not separate the data from the operating system. With the growing amount of data, data storage should be independent of operating system and can be dynamically mounted to other virtual machines. However, the traditional virtualization technology can not realize the dynamic storage of data. Docker container technology and Minio unstructured object storage technology provide a new way for us to solve the problem of mounting remote data to a virtual machine.

C. Storage technology

Traditional storage methods are block storage and file storage, but both storage methods have their own advantages and disadvantages, the emergence of object storage, a good combination of block storage and file storage advantages. Object storage introduces two concepts object and OSD, which object contains the file data and related properties and OSD is a collection of objects. Object contains four content, which is ID, data, metadata and property. ID is used to mark the object, the data is the subject of the object, the metadata used to describe the object storage and storage location, the property is defined according to the needs of other data describing the object. Objects can maintain their own property, simplifying the storage system management tasks and improving system flexibility. The size of the object can be different, the object can even contain the entire data structure, such as files, databases and so on. The bottom data of the traditional storage methods are organized in a block form, and the data underlying the data pair is found through a tree structure. However, traditional storage takes a long time to find a large number of small files. Object storage uses Hash indexing method, the direct position of data storage location, without layer-by-layer search. Object storage provides Key-Value RESTful data read and write interfaces, that often provides data access in the form of web services. Through the use of HTTP requests PUT, GET, DELETE to complete the file upload, download and delete operations. Compared with the file system, the object storage represented by AWS S3 and Swift has two notable features: a RESTful interface and a flat data organization structure. The directory tree can introduce huge spending to the storage system. The flattened data organization of the object store eliminates the need to maintain a large directory tree by dropping nested folders. The flattened data organization of the object store avoids the need to maintain a large directory tree, so lookups are fast.

Minio is an open source project by Anand Babu Periasamy. Minio is a distributed object storage server, written in Go and open sourced under Apache License Version 2.0. AWS S3

has become the standard for object storage. Minio implements AWS S3 v2/v4 Application Programming Interface. So users can access the Minio server using the Minio SDK, the Minio Client, the AWS SDK, and the AWS CLI. Minio using erasure and bitrot detection to prevent hardware failure. User can set Minio server to continuously mirror data between Minio and any Amazon S3 compatible server. Minio provides confidentiality, integrity and authenticity assurances for encrypted data with negligible performance overhead. The Minio server provides perfect technical support for Docker container technology. Use Minio Docker's image to quickly launch the AWS S3 compatible object storage server. Minio server provide object storage API access to Docker volumes. Attach Docker volumes to Minio containers and access data with REST API. Minio can be used as a cloud storage solution to save massive images, videos and documents.

Due to Minio's advantages, the platform's storage implements the way Docker deploys Minio. When multiple clients exist, each user needs to turn on a Minio service, which is inconvenient for platform management. Therefore, the docker technology is used to deploy the Minio service and docker is used to manage the Minio container to implement the multi-tenant multi-Bucket mode.

III. PLATFORM MANAGEMENT MODEL AND ARCHITECTURE

A. Platform management architecture

B. Alone-machine mode

In alone-machine mode, the tenant deploys the application in a virtual machine and the following two problems may occur while using the virtual machine.

- If a virtual machine fails, the files in the virtual machine will not be available and will continue to be used.
- Because the virtual machine's hard disk space does not change, if the data in the virtual machine becomes more and more, it may cause the virtual machine can not be used normally.

In order to solve the above two problems, this paper presents a solution. In alone-machine mode, only one tenant is using virtual machine services, so use docker container technology to open a Minio service for the tenant. This will be separated from the data and code in the application, improved data security and portability. Even if the virtual server crashes for other reasons, it will not lose data. This solution also solves the problem of virtual machine crashes caused by too much data volume, so that tenants need not worry about the shortage of virtual hard disk space. Provides the tenant with a storage policy that can be dynamically expanded and separated from the virtual server. When tenants want to share data with other servers, they only need to use the interface provided by the Minio service and do not need to perform operations such as copy and paste, thus saving computing resources and improving work efficiency.

C. Multi-machine mode

Similar to alone-machine mode, tenants deploy applications to virtual servers in multi-machine mode. Unlike alone-machine mode, multiple users share the same service in multi-machine mode. As multiple tenants use the service together, there are two problems below.

- Multiple tenants using virtual services will open multiple Minio service processes, how to manage Minio service processes on the server. If the Minio service process is not properly managed, it will cause the tenant's virtual service to fail or even crash.
- When a user frequently reads large amounts of data, the file I / O is too large and the bandwidth resource is occupied, and the server or intermediate network may be crashed and the tenant's virtual machine will not work normally.

This article presents a solution to the above two problems. In multi-tenancy mode, the platform needs to provide one Minio service for each tenant, which means that multiple Minio service processes need to be run on the server. The management of a large number of Minio server processes is a burden on the server, reducing server resource utilization. So we used Docker container technology to deploy the Minio service. Because Docker comes with its own container management technology, the Minio service can be easily managed using the Docker API. This will not be due to improper management of the Minio service process caused that the tenant's virtual server is not working properly. In order to solve the problem of the hard disk reading too fast leading to the server crashes, we added a network connection between the servers for requesting the Minio service to alleviate the bandwidth resource consumption.

Whether alone-machine mode or multi-machine mode, there are tenants have permission to share data problems. How tenants have permissions to share their own data to other tenants, and to ensure data security. Minio service can only set the file read and write permissions, but sometimes tenants just want to share part of the data to other tenants instead of all the data. In fact, each user has a Minio service. Users can create multiple buckets to store data in Minio service. After setting permissions for the pre-shared buckets, data sharing can be achieved simply and efficiently through the Minio API service.

IV. CONCLUSIONS

This management design meets our actual production needs and will be used in a variety of complex virtual machine environments such as Tensorflow and Spark in today's increasingly large data sets.

Virtual machine hard disk space will never be enhanced, only need to be able to accommodate the minimum computing environment or application resources. When deploying an application, data storage should be kept separate from the development environment so that the service delivers maximum effectiveness and security.

A. Selecting a Template (Heading 2)

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the US-letter paper size. Please do not use it for A4 paper since the margin requirements for A4 papers may be different from Letter paper size.

B. Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations

V. MATH

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

B. Units

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as 3.5-inch disk drive.
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: Wb/m² or webers per square meter, not webers/m². Spell out units when they appear in text: . . . a few henries, not . . . a few H.
- Use a zero before decimal points: 0.25, not .25. Use cm³, not cc. (bullet list)

C. Equations

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled. Number equations consecutively. Equation numbers, within parentheses, are to position flush right, as in (1), using a right tab stop. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in

$$\alpha + \beta = \chi \quad (1)$$

Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use (1), not Eq. (1) or equation (1), except at the beginning of a sentence: Equation (1) is . . .

D. Some Common Mistakes

- The word data is plural, not singular.
- The subscript for the permeability of vacuum μ_0 , and other common scientific constants, is zero with subscript formatting, not a lowercase letter o.
- In American English, commas, semi-colons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an inset, not an insert. The word alternatively is preferred to the word alternately (unless you really mean something that alternates).
- Do not use the word essentially to mean approximately or effectively.
- In your paper title, if the words that uses can accurately replace the word using, capitalize the u; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones affect and effect, complement and compliment, discreet and discrete, principal and principle.
- Do not confuse imply and infer.
- The prefix non is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the et in the Latin abbreviation et al..

- The abbreviation i.e. means that is, and the abbreviation e.g. means for example.

VI. USING THE TEMPLATE

Use this sample document as your LaTeX source file to create your document. Save this file as **root.tex**. You have to make sure to use the cls file that came with this distribution. If you use a different style file, you cannot expect to get required margins. Note also that when you are creating your out PDF file, the source file is only part of the equation. *Your \TeX \rightarrow PDF filter determines the output file size. Even if you make all the specifications to output a letter file in the source - if you filter is set to produce A4, you will only get A4 output.*

It is impossible to account for all possible situation, one would encounter using \TeX . If you are using multiple \TeX files you must make sure that the “MAIN“ source file is called root.tex - this is particularly important if your conference is using PaperPlaza’s built in \TeX to PDF conversion tool.

A. Headings, etc

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named Heading 1, Heading 2, Heading 3, and Heading 4 are prescribed.

B. Figures and Tables

Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation Fig. 1, even at the beginning of a sentence.

TABLE I
AN EXAMPLE OF A TABLE

One	Two
Three	Four

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in a document, this method is somewhat more stable than directly inserting a picture.

Fig. 1. Inductance of oscillation winding on amorphous magnetic core versus DC bias magnetic field

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when

writing Figure axis labels to avoid confusing the reader. As an example, write the quantity Magnetization, or Magnetization, M, not just M. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write Magnetization (A/m) or Magnetization A[m(1)], not just A/m. Do not label axes with a ratio of quantities and units. For example, write Temperature (K), not Temperature/K.

VII. CONCLUSIONS

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

APPENDIX

Appendixes should appear before the acknowledgment.

ACKNOWLEDGMENT

The preferred spelling of the word acknowledgment in America is without an e after the g. Avoid the stilted expression, One of us (R. B. G.) thanks . . . Instead, try R. B. G. thanks. Put sponsor acknowledgments in the unnumbered footnote on the first page.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

REFERENCES

- [1] W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123135.
- [2] W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123135.
- [3] H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.
- [4] B. Smith, An approach to graphs of linear forms (Unpublished work style), unpublished.
- [5] E. H. Miller, A note on reflector arrays (Periodical styleAccepted for publication), IEEE Trans. Antennas Propagat., to be published.
- [6] J. Wang, Fundamentals of erbium-doped fiber amplifiers arrays (Periodical styleSubmitted for publication), IEEE J. Quantum Electron., submitted for publication.
- [7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.
- [8] Y. Yoroazu, M. Hirano, K. Oka, and Y. Tagawa, Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style), IEEE Transl. J. Magn.Jpn., vol. 2, Aug. 1987, pp. 740741 [Dig. 9th Annu. Conf. Magnetics Japan, 1982, p. 301].
- [9] M. Young, The Techincal Writers Handbook. Mill Valley, CA: University Science, 1989.
- [10] J. U. Duncombe, Infrared navigationPart I: An assessment of feasibility (Periodical style), IEEE Trans. Electron Devices, vol. ED-11, pp. 3439, Jan. 1959.
- [11] S. Chen, B. Mulgrew, and P. M. Grant, A clustering technique for digital communications channel equalization using radial basis function networks, IEEE Trans. Neural Networks, vol. 4, pp. 570578, July 1993.
- [12] R. W. Lucky, Automatic equalization for digital communication, Bell Syst. Tech. J., vol. 44, no. 4, pp. 547588, Apr. 1965.
- [13] S. P. Bingulac, On the compatibility of adaptive controllers (Published Conference Proceedings style), in Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory, New York, 1994, pp. 816.
- [14] G. R. Faulhaber, Design of service systems with priority reservation, in Conf. Rec. 1995 IEEE Int. Conf. Communications, pp. 38.

- [15] W. D. Doyle, Magnetization reversal in films with biaxial anisotropy, in 1987 Proc. INTERMAG Conf., pp. 2.2-12.2-6.
- [16] G. W. Juetten and L. E. Zeffanella, Radio noise currents in short sections on bundle conductors (Presented Conference Paper style), presented at the IEEE Summer power Meeting, Dallas, TX, June 22-27, 1990, Paper 90 SM 690-0 PWRS.
- [17] J. G. Kreifeldt, An analysis of surface-detected EMG as an amplitude-modulated noise, presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.
- [18] J. Williams, Narrow-band analyzer (Thesis or Dissertation style), Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.
- [19] N. Kawasaki, Parametric study of thermal and chemical nonequilibrium nozzle flow, M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.
- [20] J. P. Wilkinson, Nonlinear resonant circuit devices (Patent style), U.S. Patent 3 624 12, July 16, 1990.