

· 信息技术与通信 ·

# 基于 Shell 脚本的 KVM 虚拟化技术应用

许志聪

(广东工程职业技术学院, 广东 广州 510520)

**【摘要】**虚拟化技术是当前计算机信息技术中一个新的变革,其中开源的虚拟化技术(Kernel-Based Virtual Machines, KVM)更是以稳定、高效、经济而著称,为科学研究、工程仿真等领域提供了强大的计算能力。文章以某高职院校虚拟化应用为实例,探讨在普通高职院校的虚拟化技术应用管理中使用 Shell 脚本进行智能化管理,并展示其技术用于云平台辅助管理的优势,为用户轻便智能管理 KVM 虚拟化技术应用提供参考。

**【关键词】**虚拟化; KVM; Shell 脚本

**【中图分类号】** TP311

**【文献标识码】** A

**【文章编号】** 1008-1151 (2024) 03-0009-04

## 0 引言

开源的虚拟化技术(Kernel-Based Virtual Machines, KVM)是基于虚拟化扩展的 x86 硬件的开源 Linux 原生的全虚拟化方案,在 2.6.20 以上 Linux 的内核中均已包含了 KVM 核心,是目前流行的虚拟化产品,具有开源的特性<sup>[1]</sup>。在当前国内外主流云计算平台及其虚拟化应用中,KVM 技术占据主导地位。KVM 虚拟化技术可以实现资源的有效利用,通过虚拟化技术,可以在单个物理服务器上运行多个虚拟机,从而充分利用服务器的硬件资源,提高资源的利用率。KVM 虚拟化技术的研究促进了高性能计算的发展。KVM 虚拟化技术可用于创建和管理高性能计算集群,支持大规模并行计算任务,为科学研究、工程仿真等领域提供了强大的计算能力。本文主要探讨 Shell 脚本在 KVM 虚拟化中的应用,为读者提供一种轻便智能管理 KVM 虚拟化的思路。

## 1 虚拟化技术

虚拟化是指通过虚拟化技术将一台计算机虚拟为多台逻辑计算机。在一台计算机上可以同时运行多台逻辑计算机,每台逻辑计算机可以运行不同的操作系统<sup>[2]</sup>。通过虚拟化技术,系统管理员可以动态

地分配服务器的资源,从而提高硬件资源的使用率。当前,随着云计算、大数据、区块链等技术的应用推广,虚拟化技术应用更加广泛。

## 2 KVM 技术

### 2.1 KVM 虚拟化应用

虚拟化技术主要分为硬件及系统虚拟化、应用软件及进程虚拟化,而对于硬件及系统虚拟化,VMware 和 KVM 是应用最为广泛的。其中 VMware 是 VMware 公司开发的专有虚拟化技术,提供的虚拟化技术及解决方案需要较高的费用。

KVM 是基于 Linux 系统的一个开源的虚拟化模块,能够模拟出主机中的各个硬件信息,被广泛应用于各类服务器虚拟化、客户端桌面虚拟化、存储及网络虚拟化等。由于各商业公司在使用开源的 KVM 虚拟化技术时,是不需要支付 KVM 虚拟化相关软件费用的,因此大量的中小公司、学校等都使用 KVM 的底层虚拟化技术作为其商业化应用的技术解决方案。

### 2.2 KVM 管理工具和方法

KVM 的管理模式有以下 3 种方法:

(1) 使用 KVM 内置的图形化管理工具 virt-manager。

**【收稿日期】** 2023-06-27

**【基金项目】** 广东工程职业技术学院 2021 校级科研项目(KYYB2021018)。

**【作者简介】** 许志聪(1981—),男,讲师,计算机科学实验师,硕士,研究方向为计算机网络工程技术、云计算应用。

(2) 使用 KVM 内置的 `virsh` 非交互式命令进行管理；使用 `virsh` 命令结合 KVM 虚拟机的配置文件 `xml` 编辑可以对 KVM 虚拟化进行各类管理和配置。

(3) 使用云主机平台进行统一化管理，这类平台包括第三方开源的 `openstack` 以及各个厂商自行研发的云管理平台，例如华三的超融合管理平台 H3C UIS、锐捷的云管理平台 RG-JCOS 等。这类管理平台虽然界面友好、操作简单，但是存在运行效率较低、有排他现象等缺点。

### 2.3 Shell 脚本自动化管理 KVM

Shell 作为人与操作系统的一个接口，shell 脚本类似编程里面的解释程序，有自己的一套命令集<sup>[3]</sup>，也可以被其他程序调用，自身也包含了编程所需的各类语句、变量、数组、自定义函数等基础语法结构，是 linux 系统下的一个很好的操作工具。在 KVM 服务器中使用 shell 脚本管理 KVM 虚拟机是一种成本较低、效率较高的简易模式。Shell 脚本能够直接调用 KVM 中的 `virsh` 工具，并通过内置文本处理工具 `awk`、`sed` 等对 KVM 中的虚拟机各类配置文件 `xml` 进行标准化、定制化编辑，从而有效发挥管理 KVM 虚拟主机的作用。

## 3 KVM 管理脚本开发

KVM 的 `virsh` 工具涵盖了主机域管理、系统信息监控、主机及管理系统、网络接口、网络模式、设备管理、快照管理、存储管理、存储池管理、系统性能监控和系统迁移等模块。系统管理脚本可以根据 `virsh` 工具所提供的各个功能模块进行定制化的编写，并根据每个服务器实际应用场景及常用功能进行细化。本文将挑选 4 个常见的模块功能进行举例阐述。

### 3.1 KVM 虚拟机基础管理

KVM 虚拟机基础管理主要包含 KVM 主机信息、虚拟主机的镜像管理情况、主机配置 XML 文件的目录及内容查询、主机基础信息等内容。

通过模块函数实现的 KVM 服务器不仅可以自动查看所有主机、镜像文件在服务器上的绝对路径，还可以查看每个 KVM 主机的系统信息。用户还可以根据平时的管理习惯进行有效的扩展，例如查看虚拟机的内存、硬盘容量及剩余容量；查看 CPU 的个数、内存容量及可用容量等。此外，还可以进行

修改内存容量和 CPU 个数，增加硬盘、网卡和光驱等常规操作。

### 3.2 KVM 虚拟主机自动化克隆

Redhat 公司系列下的 Linux 操作系统在 6.0 之前使用的镜像都是 `raw` 格式，后缀名通常是 `img`，这种文件格式最大的缺点是不支持快照技术。而当前 KVM 虚拟化技术使用的文件系统默认格式是 `qcow2`，这个格式大部分性能与传统的 `raw` 相差无几，主要是增加了支持快照这一功能。KVM 的管理服务器把 KVM 虚拟机视为一个整体的文件进行管理，因此可以实现高效且简易的管理。理论上每新建一个 KVM 虚拟主机，就会多出一个 `qcow2` 镜像文件，当物理主机存储空间出现瓶颈时，可以通过 NFS 等网络服务共享镜像的存储，从而使虚拟主机的克隆变得更加方便，只需要为虚拟主机配置合适的 XML 文件即可。在相同的 KVM 服务器下，不同的 KVM 虚拟主机系统对应的 XML 文件中有以下 3 个参数是不能够重复：UID、主机名和 MAC 地址。本文通过 Shell 脚本自动化完成了一个完整的克隆函数模块编写。虽然 KVM 的图形管理界面 `virt-manager` 也提供了克隆的功能，但是该操作过程需要源虚拟化主机系统处于关闭状态，且图形界面无法完成批量化操作，因此使用脚本进行克隆才是当前的主流操作方法。

自定义函数将与克隆的新机器系统相关的必选参数定义为变量，这样只需要调用 `bash shell` 脚本的 `for` 循环或者 `until` 循环语句，服务器管理员就可以按照命令批量化、无人值守地完成系统克隆。例如需要批量增加一批主机，这些主机分别是 `test01`、`test02`、...、`test50`，只需要在 Shell 下调用常规 `for` 循环代码即可完成该任务。

### 3.3 KVM 快照管理

KVM 快照是存储快照技术的一种应用，快照技术在当前计算机应用中非常广泛，特别是在服务器的应用中更是如此。它能够让操作系统数据迅速恢复到指定的某个时间点，并且可以使用很小的代价完成较多的系统数据备份和系统容灾恢复。

KVM 快照有多种实现模式和技术手段，包括内存快照、内置快照和外置快照。其中，内存快照可以对系统当前内存状态下的硬盘数据进行一次完整的备份，生成一个新的镜像压缩文件。虽然这种快

照模式没有限制快照次数，但是对硬盘容量的要求较高。每次快照都会生成一个类似于备份的镜像文件，该文件的容量约是原镜像的 1/3。每当系统恢复回滚时，只需要指定该备份文件并进行相关的命令操作即可。然而，如果快照次数较多，硬盘容量增加会更快，这对服务器的硬盘压力较大。

相比之下，内置快照更加符合大多数应用场景和成本核算的要求，内置快照每次快照后会将快照相关的信息和数据等附加在系统镜像 qcow2 的文件上，因此每次完成快照的制作后，系统镜像容量都会变大。一个 qcow2 格式的镜像支持 20 个内置快照，这对于服务器系统来说已经足够。KVM 的图形管理工具 virt-manager 也支持内置快照。因为内置快照支持虚拟机系统各类状态（运行或关闭）且操作简单，qemu 中自带的工具 virsh 也支持快照制作和快照回滚，所以使用内置快照的管理员比例相对较高，其应用场景也较为广泛。

外置快照每次在创建时会把当时系统的状态保存在磁盘的一个 backing file 文件中，快照产生后的磁盘数据更新 overlay 则写在一个新的快照文件里面。在这种情况下，每次快照后产生的数据相对可控，对磁盘的压力也相对较小。但是 KVM 中的 qemu 技术并不支持外部快照系统回滚，系统回滚过程需要对底层配置文件中的镜像信息进行手动修改，回滚操作步骤比较烦琐且不允许 KVM 虚拟机处于开机状态，所以在服务器端的虚拟化应用相对较少。使用外部快照更多的是桌面虚拟化，主要应用于增

量数据的克隆、下发等功能。许多商业化的云桌面管理平台通过二次开发，将烦琐的操作过程编译成图形化菜单及按钮，大量地应用 KVM 外置快照这一技术。

同理，作为 KVM 的常见功能，快照这一部分功能也可以使用对应的 Shell 脚本进行模块化管理。与快照相关的脚本模块编写具有自定义快照的名称、显示快照所有列表，以及删除快照、快照回滚等功能，能够通过一个程序的运行实现快照管理所有功能。该操作整体而言较为直观且快捷。

### 3.4 KVM 系统迁移

每当虚拟机的物理服务器的资源或使用年限达到某个阈值，或服务器访问量剧增时，KVM 虚拟机系统可能需要完成部分或者整体迁移。虽然这个操作不是特别常用，但却是 KVM 虚拟化服务中非常重要的一环。

系统迁移的脚本主要通过理解 KVM 虚拟化的技术原理，完成对 KVM 虚拟化的底层文件管理，virsh migrate 模块和图形管理界面 virt-manager 是没有这项技术的。相较于 virsh 工具中的 migrate 模块，KVM 虚拟化技术更为成熟和可靠。

笔者通过 KVM 基础管理、系统克隆、快照应用这 3 个模块的内容，对 virt-manager 和 virsh 命令及 Shell 脚本应用进行了操作复杂度和操作及运行时间的统计对比，发现使用脚本完成各个单项工作的优势明显，结果如表 1 所示。

表 1 不同工具效率对比表

单项工作	基础管理	系统克隆	快照应用
virt-manager 图形操作步骤	大约 10 次按钮点击	大约 2 次按钮点击（无法批量）	大约 3 次按钮点击（灵活性差）
virsh 命令操作步骤	大约 5 个步骤	大约 6 个步骤	大约 5 个步骤
Shell 脚本操作步骤	1 个步骤	1 个步骤	1 个步骤
virt-manager 图形操作消耗步骤	约 3 min	3G 镜像约 5 min	3G 镜像约 20 s
命令模式工作	约 2 min	3G 镜像 6~10 min	3G 镜像约 2 min
Shell 脚本模式工作	10 s 以内	3G 镜像约 3 min	3G 镜像 20 s 以内

## 4 基于 KVM 的 shell 脚本小结

### 4.1 在 KVM 应用范围

本文仅通过 KVM 基础管理、主机克隆、快照应用、系统迁移进行实验，并与传统的图形管理模式及 virsh 命令模式进行对比。实际上，在 KVM 下

的任意操作都可以通过 Shell 脚本完成，例如安装部署 KVM 及其相关工具，自动安装 KVM 虚拟机系统，镜像压缩检测，KVM 网络模式管理及自动化切换，内存快照、外部快照管理及应用，镜像存储池管理，网络存储池自动添加等。

（下转第 20 页）

的前提下,提供了多样化的部署方式,并对其进行了全面的测试和验证,以确保选择的部署方式能够满足不同用户的需求。因此,开发者和用户在使用过程中,可以根据实际情况灵活地选择最适合的部署方式。部署服务器需要满足的最低要求:

(1) 全系统的操作系统,最小版本为 7.3。

(2) 容器为 docker,最小主版本为 19。实体关系规模为十万级以下的,建议部署 16 核、32G 服务器 1 台;百万级以下十万级以上的建议部署 32 核、64G 服务器 1 台;千万级的建议部署 32 核、64G 服务器 5 台。

## 5 结论

文章提出的元宇宙多人交互引擎,为元宇宙用户提供了开发元宇宙产品的后端技术支撑能力。用户只需通过 SDK 的 API 调用即可完成平台对接,获得开发功能完整的元宇宙产品的能力。平台将多人同屏交互同步、优化流媒体传输等技术运用到元宇宙产品开发中,并将其封装为成熟技术供第三方开发者使用。平台通过开发元宇宙模型资产库,不仅为第三方开发者提供 500 种以上高清 3D 模型,

还提供 U3D 和 UE 的 SDK 快速接入,支持 AR、VR、PC、移动端同屏互动。自研 Websocket 长连接网关,支持各种元宇宙应用场景下对消息数据的路由分发。元宇宙沉浸式多人交互平台的核心价值是为高校师生及初创公司赋能,让更多的人可以低成本且快速地进行元宇宙创作。通过 SDK 接入平台,获得后端平台能力支撑。企业无需专职的后端开发人员、跳过后端功能实现和前后端沟通对接等环节,节约了人力和时间成本,可以集中精力进行元宇宙产品的前端内容制作和产品创新。

## 【参考文献】

- [1] 彭影彤,高爽,尤可可,等.元宇宙人机融合形态与交互模型分析[J].西安交通大学学报(社会科学版),2023,43(2):176-184.
- [2] 任天.基于交互叙事的校园文化游戏化体验 APP 设计与实现:以北邮 AR 元宇宙为例[D].北京:北京邮电大学,2022:55.
- [3] 郭亚军,袁一鸣,张腾飞.元宇宙场域下用户信息交互生态机制研究[J].农业图书情报学报,2022,34(6):4-13.

(上接第 11 页)

## 4.2 管理平台集成开发

目前普通用户使用的 KVM 虚拟化技术主要应用于私有云服务,各个私有云通常都有自建或厂家提供的云管理平台。通常这类平台的菜单、功能和按钮都是固定的,通过 Shell 脚本也可以辅助完成部分自动化管理操作。一旦网络管理员需要增加一些管理平台没有提供的工具,需要在平台上进行二次开发,所需的时间和人力成本是比较高的,而在大部分人力和资金相对有限的单位,这很可能无法实现。在这种情况下,使用基于 shell 的 KVM 虚拟机管理脚本来辅助管理是更为合理且有效的。

## 5 结论

本文旨在简要阐述 Shell 脚本在 Linux 操作系统中,特别是在涉及开源虚拟化技术 KVM 的管理

应用时,所发挥的作用和优势。根据笔者的实践应用经验,可以尝试将这部分知识作为日常服务器管理的辅助工具,以提高日常网络及服务器管理的效率,节约部分人力成本。大中院校的实践教学也可以将这些知识点应用到课程教学中,将学习的知识直接应用到实践中,从而增强学生的专业技术信心和动力。

## 【参考文献】

- [1] 梁志荣. KVM 技术的多媒体课堂管理[J].信息与电脑(理论版),2021(9):197-199.
- [2] 王金恒,刘卓华. KVM+Docker+Openstack 实战:虚拟化与云计算配置、管理与运维微课视频版[M].北京:清华大学出版社,2021(1):3.
- [3] 刘佳,刘祖耀. Shell 脚本在云服务器管理中的应用[J].计算机产品与流通,2017(12):51.