基于 Docker 的应用软件虚拟化研究

马　越

南京邮电大学 计算机学院，江苏 南京 210023

摘 要：应用软件虚拟化技术是云计算虚拟化中的重要内容，是对传统的基于 B/S 模式的 SaaS 的一种补充，简化了云平台下软件部署的方式，优化了云平台的管理。本文探讨现有的应用软件虚拟化技术，分析比较现有的应用虚拟化的解决方案，并提出了新的一套基于 Docker 技术和 VNC 协议的应用软件虚拟化方案，并通过实例应用展示，说明了基于 Docker 技术的应用软件虚拟化技术具有跨平台性及快速部署的优势，能够在一定程度上替代现有的一些应用软件虚拟化技术。

关键字：XenApp；ThinApp；Docker；VNC

Research on Application Virtualization Based on Docker

MA Yue

College of Computer Science & Technology, Nanjing Univ. of Posts and Telecommunications, Nanjing 210023 , China

Abstract: Application software virtualization technology is important content in cloud computing software, is a complement to the B/S model based on the traditional SaaS, simplifies the deployment of software under the cloud platform,optimize the cloud platform management. This paper explores the exist application software virtualization technology, analyse and compare the solutions of the existing application virtualization, and puts forward a new solution of application software virtualization based on the Docker technology and VNC protocol, and through examples show, to prove that the application software virtualiztion technology based on Docker technology is cross platform and fast deployment, it can replace some application software virtualization technology existing in a certain extent.

Keywords: XenApp；ThinApp；Docker；VNC

1 引言

伴随着云计算技术的发展，软件即服务（Software as a Service，SaaS）得到快速发展，据 Gartner 估计，到 2015 年全球软件即服务的收入将会达到 213 亿美元[1]。然而，传统的软件即服务采用的是 B/S 的架构。通过浏览器进行交互虽然能够免除在终端安装客户端，且减少对于跨平台的客户端软件开发，具有更好的适应性；但这种瘦客户端的方式还是存在一些限制的，且要把现有的众多客户端应用软件完全替换掉，其过程将会是十分缓慢的。为了弥补浏览器的不足，一些厂商提出了一种直接把现有的应用软件进行虚拟化部署的技术，主要产品为 Citrix 的 XenApp 和 Vmware 的 ThinApp。这两款产品的主要功能就是将应用程序打包成不需要安装即可运行的单一可执行程序，让云桌面的系统直接运行该程序而不需要安装。然而，这两款产品都属于商业性质的产品，除了需要购买该产品外，还必须是使用其自己厂商的虚拟化软件，所以还是具有一定的局限性。然而，目前还没有一种通用的解决方案。

2 现有的应用虚拟化技术

应用虚拟化技术随着 SaaS 的发展而不断发展，但其本身并没有一个被广泛认可的定义[2]，但其核心都是把操作系统与应用软件运行环境相分离，可以简单的理解为：将应用程序的应用界面与实际应用运行分离，通过在服务器上为用户开设独立的会话，占用独立的内存空间，应用程序的计算逻辑指令在这个会话空间中运行，应用程序的界面通过协议传送到用户终端，这样用户通过网络可以高效、安全地访问服务器上的各种应用软件，获得在本地运行应用一样的体验[3]。

当然，对于类似于支持 Windows 软件运行于 Linux 上的开源软件 Wine[4]，以及支持跨 Linux 发行版本运行的开源软件 CDE[5]，这些软件虽然都是在本地运行，但其也是实现了应用程序与操作系统相隔离，所以也都属于应用软件虚拟化技术。这种支持应用软件运行在异构平台上运行的虚拟化技术，其缺陷比较明显，即需要消耗本地计算机的资源来实现虚拟化，且因为是异构平台，受到运行环境的限制，并不能很好的实现虚拟化，技术不够成熟，虽然有商业化的版本，但目前并没有广泛的应用。

所以，针对当前的应用虚拟化技术，我们大致可以分为四类：

1.同构本地应用虚拟化：在本地同架构的环境下实现应用程序与操作系统的分离，代表产品位 Linux 下的 CDE；

2.同构远程应用虚拟化：在与客户端同架构的环境下，把应用程序通过服务的形式给客户端进行

基金项目：江苏省高校教育学会基金（JSSY1201），南京邮电大学基金（SG1107）

作者简介：马越（1990～），男，汉族，江苏南京人，硕士研究生。

使用，代表产品为 VMware 的 ThinApp；

3.异构本地应用虚拟化：在本地异构平台的环境下实现应用程序的虚拟化，代表产品为 Wine；

4.异构远程应用虚拟化：应用软件在与客户端的不同架构环境下，以流的方式推送到客户端，此类的代表产品为 Citrix 的 XenApp；

目前得到较多应用的应用软件虚拟化技术主要是由几大厂商推动的商业产品:Citrix 的 XenApp，VMware 的 ThinApp 以及 Microsoft 的 App-V，这些应用软件虚拟化解决方案都是服务于其各大公司的虚拟化云桌面的。而由于目前市场上虚拟化云平台应用的较多的是 VMware 和 Citrix 的解决方案，所以 XenApp 和 ThinApp 更为常见。

Citrix 的 XenApp 主要是通过 ICA 协议，该协议连接了运行在服务器上的 XenApp 的应用软件进程和客户端的设备，正是通过 ICA 的虚拟通道，使得应用客户端软件不安装在客户端设备上也能使得用户感觉就如在本地使用一样[6]。对于这种将应用集中到数据中心，并以按需服务方式交付这些应用，Citrix 的 XenApp 因为较早投入市场，所以现在被认为是这类软件中的事实上的标准了。

VMware 的 ThinApp，是 VMawre 的云桌面产品 VMware View 的重要组件之一，以消除应用程序冲突和简化管理为目标而设计，可简化应用程序虚拟化，并可降低应用程序交付的成本或复杂性。ThinApp 是一种无代理的应用程序虚拟化解决方案，可将应用程序与其底层操作系统剥离开来，以消除应用程序冲突，并简化应用程序的交付和管理。ThinApp 是 VMware 收购 Thinstall 后推出的产品，以应对 Citrix 的 XenApp。

近年来，由于信息技术产业的高速发展，许多企事业单位为了更好的管理计算机设备，便通过引入桌面云系统，来进行计算机设备的管理。例如，南开大学医学院[7]、哈尔滨工业大学图书馆[8]、宁波市气象服务中心[9]，都是采用的 VMware 的桌面云解决方案，即使用 ThinApp 来进行软件虚拟化，而如福州大学网络中心[10]则是使用的 Citrix 的产品。可以说，对于新部署的桌面云环境来说，为了减少云桌面的负载，一般都会采用应用软件虚拟化技术，其次，因为出于对信息安全的考虑，使用应用虚拟化产品能够更好的保护系统的安全，例如在保证学校机房的信息安全方面[11]，应用虚拟化能够极大的提高信息系统环境的安全性。所以说，应用软件虚拟化正随着云计算的发展而得到广泛的应用。

3 基于 Docker 构建应用虚拟化

现有的应用虚拟化技术虽然在性能上已经满足了企业的一般性需求，但其普遍存在一个重要的问题，就是对底层依赖较强，不能够独立使用。例如 ThinApp，是作为 VMware View 的组件，所以要想使用 ThinApp，则需要投入较多的资金，且在 VMware 下的应用虚拟化，并不能很好的提供给其它云桌面的环境。使用传统的方法不能够很好地解决该问题，但近年来新出现的 Docker 技术则给我们提供了一个解决该问题的方向。

Docker 技术是基于容器技术的一种实现，而容器技术是一种轻量级的虚拟化技术，容器有效地将由单个操作系统管理的资源划分到孤立的组中，以便更好地在孤立的组之间平衡有冲突的资源使用需求[12]。所谓轻量级，是因为容器的思想就是把单独的进程进行封装，在不影响系统内的程序的情况下运行。这样做就使得容器内的进程所运行的环境是与其它任何操作系统相独立。这一点符合应用软件虚拟化技术所要求的把操作系统与应用软件运行环境相分离，所以可以说容器技术十分接近于应用软件虚拟化技术。但容器技术与应用虚拟化技术还是有所区别的，容器技术所依赖的是其运行底层的操作系统的内核，所以其并没有完全实现与操作系统的解耦合。但容器技术通过独立的命名空间，保证了其运行的独立性。命名空间(Namespace)是 Linux 的内核针对实现容器虚拟化而引入的一个强大特性，实现了对内存、CPU、网络 IO、存储空间、文件系统、网络、PID、UID、IPC 等等的相互隔离，所以通过容器运行应用软件，能够使得应用程序与操作系统实现极低的耦合，如果考虑其使用内核是通过封装的接口，且其依赖内核只是因为其为了做到轻量，才采取共用内核的措施，其和本地系统在内核上已实现了相互隔离，我们可以认为其在一定程度上就已经实现了与操作系统的解耦合。所以，我们可以通过容器技术来构建应用虚拟化，由于容器启动都在本地，所以可以算是同构本地应用虚拟化。但因为目前云桌面系统都是希望能够实现远程访问的方式，这也是 SaaS 的发展目标，所以，为了达到这一需求，我们需要进行改进。

从对现有虚拟化应用技术的分析中，我们可以看出，要实现异构远程应用虚拟化，需要服务端的应用程序以流的方式推送到异构的客户端。所以，我们采用在 Docker 中增加 VNC 的方式来实现应用程序与客户端的流数据传输。虚拟网络计算(Virtual Network Computing,VNC)是一种优秀的开源轻量型的远程计算机控制软件。它最早是由英国剑桥大学 AT&T 实验室设计并开发,使用了 GPL 授权，任何个人或机构都可以免费获取该软件[13]。该软件经过多年的发展，已经有了各个平台下的客户端版本，所以，可以说在任意平台下都可以通过 VNC 客户端软件访问 VNC 的服务端程序。正是基于这种广泛的应用基础，所以我们可以使用 VNC 来访问我们的虚拟化的应用。文献[14]中以 VNC 为基础，通

过在 KVM 的解决方案中集成 VNC Server，在多平台下通过 VNC 客户端软件对云平台下的 KVM 虚拟机进行管理。基于上述思想，我们实现了在 Docker 创建的容器中添加 VNC Server，并通过修改 VNC Server，让其把 Docker 容器所封装的应用发布出去，让用户可以直接在任何平台下通过 VNC 客户端软件直接访问应用程序，而不需要在客户端上安装对应的应用软件，这样达到了 XenApp 和 ThinApp 的效果。且由于 Docker 本身是可以在多个平台上安装的，所以不需要像 XenApp 和 ThinApp 一样，只能适用于自己厂商所支持的云平台环境。且由于 Docker 和 VNC 的客户端软件都是开源产品，可以使用免费版本，所以就不需要花费大量金钱去购买这些应用软件虚拟化产品，这样可以为企业减少部署的成本，从而提高投资回报率。

我们构建的架构环境如图 1 所示：

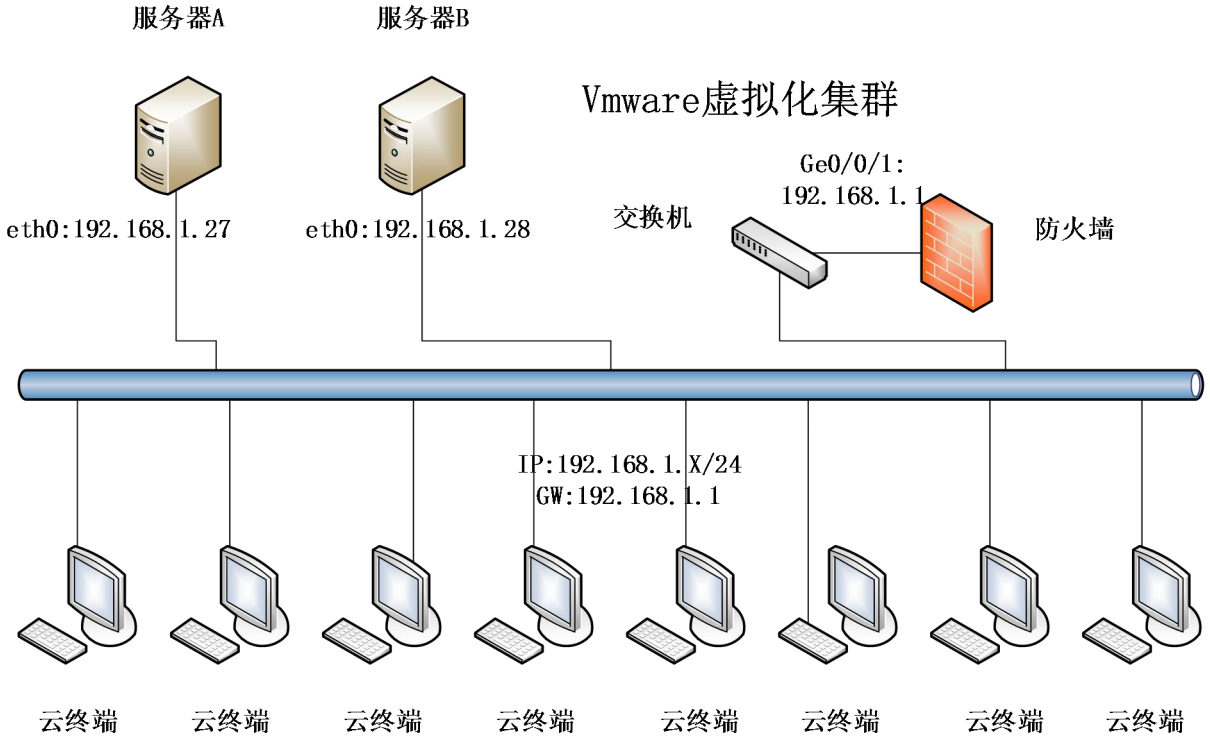

图 1 实验部署环境拓扑图

我们在服务器 A 上部署了 VMware 的 Vsphere5.5，部署了八个云桌面系统，提供给八个云终端使用，在服务器 B 上，则是搭建了 Docker 的环境，主要用来提供应用软件虚拟化。各云终端中使用的都是 Windows 的云桌面，我们使用 RealVNC 的客户端来连接我们的虚拟化应用。

对于应用软件的虚拟化，我们是通过 Docker 把其封装成对应的应用软件镜像，以便于随时取用。创建 Docker 镜像有两种方法，创建 Docker 的第一种方法使用 docker commit 命令，第二种方法就是是使用 Dockerfile 文件和 docker build 命令来构建镜像[15]。Docker 官方更推荐使用用 Dockerfile 文件的方式进行构建镜像，因为这样便于大家自己进行构建，我们也主要以该方法构件镜像，并把所写的 Dockerfile 及所需文件上传到 Github 上进行共享。

通过在我们的虚拟化实验室中进行部署该软件虚拟化技术，使我们的用户在云桌面环境下能够直接通过 VNC 使用类似 Firefox 之类的图形化的软件，这样系统管理员就不需要为在云桌面上安装软件而修改系统模板，有利于减少系统管理员的工作，从而能够降低出错几率。

4 性能评价

应用软件虚拟化的性能评价目前来说缺乏一个统一的标准，通过分析目前市场上获得用户认可的应用虚拟化软件，我们从下面几个方面来评价应用虚拟化的性能：

启动时间：应用程序的启动时间是衡量所有软件的一个重要指标，是体现软件性能优劣的重要参考，虚拟化的应用软件亦是如此，只有启动快速的虚拟化应用才能获得用户的认可，才能得到推广；

隔离度：应用软件虚拟化应当与运行的系统高度隔离，互不影响，这是所有软件应用虚拟化所必需的；

平均无故障时间：应用软件是否可靠，能否商用，平均无故障时间是衡量一个软件稳定性的重要指标，对于应用虚拟化软件来说，影响其平均无故障的时间较多，难以给出一个完全的测试评价；

安全性：对于应用软件虚拟化来说，其本身的一大特点就是通过与本地系统隔离，来提高安全性，因此，应用虚拟化软件的安全性是十分重要的一项参数；

速度比：主要是指采用应用虚拟化的软件与在本地使用在执行统一任务时的比值，该值能够体现应用虚拟化软件的性能，该值越低越好；

我们所构建的基于 Docker 的应用软件虚拟化方案，在启动时间上是拥有极大的优势的，这是因为我们把应用封装在 Docker 容器中，Docker 的容器启动时间是很快的，因此，在网络环境条件允许的情况下，基于 Docker 的应用软件虚拟化所提供的软件在启动时间上是不逊色于本地软件的，有些情况下甚至优于本地软件，例如我们在使用 Firefox 软件时，本地初始启动是比较缓慢的，而我们通过 VNC 连接，其启动速度是很快的。这是因为本地启动 Firefox，在初始加载时需要消耗本机较多的内存来进行加载，在本机运行程序较多的时候会使该过程比较缓慢，而我们通过 VNC 连接则不存在这些问题，所以在速度上表现优异；也因为同样的原因，速度比也是较低的，且根据我们实际测试，在启动 Docker 容器时，其对内存的占用是相当少的，例如我们通过比较在启动 Firefox 镜像前后的系统内存占用发现，其内存总共消耗不到 50M，可见其对资源的占用是多么低的。对于隔离度上的表现，由于本身 Docker 是使用的容器技术，该技术的隔离性已经经过了大量的检验，所以在隔离度上是不逊色于其它应用软件虚拟化技术的。而且，由于有高度的隔离性，所以该应用虚拟化也能保证较高的安全性，即使虚拟化的应用软件受到威胁，只需重新启动一个新的容器即可，不会受到任何持续性的影响。使用 Docker 进行软件虚拟化，其在应用损坏后只需重新在 Docker 上在启动一个封装该应用的镜像，即可再次正常使用，而 Docker 启动又是十分迅速的。

综上所述，使用基于 Docker 的软件虚拟化，能够在性能上满足一般的应用虚拟化的需求，因此，我们的这套架构方案是能够适用于一般的软件虚拟化环境，能够部分替换现有的诸如 XenApp 之类的应用虚拟化产品，从而节约软件使用成本。

5 结束语

在我们实际部署的环境中，基于 Docker 构建的应用虚拟化软件能够流畅的使用，取得了良好的效果，为我们对云平台的管理提供了很大的帮助。当然，现阶段使用 Docker 来进行软件虚拟化还存在一定的局限性，即只能虚拟 Linux 下的应用软件，对于 Windows 下的软件则无法进行软件虚拟化。但是由于 Docker 技术在高速发展，微软也在 2014 年 10 月份表示会开始考虑在 Windows 上支持 Docker 技术，对于在 Docker 下构件应用虚拟化也会有所提高。当然，Docker 的发展还不仅限于此，Docker 先阶段以表现出了其强大的虚拟化优势，在云计算领域得到了许多大厂商的支持，例如 RedHat、AWS 等云计算相关的企业，国内目前也在一些社区的推动下开始兴起对 Docker 的研究，所以说 Docker 未来的发展还是比较广阔的。

参考文献

[1] Wikipedia.Software as a service[EB/OL].(2012-05-24)[2015-1-7].http://en.wikipedia.org/wiki/Software-as-a-Service.

[2] 陈靖，黄聪会，孙璐，龚水清.应用虚拟化技术研究进展[J].空军工程大学学报，2013，14(6)：54-58.

[3] 付平武，幸筱流.VMware 虚拟化技术在教学机房的应用[J].电脑知识与技术，2013，9(30)：6864-6866.

[4] 王亚军,刘金刚.Windows 程序运行于 Linux 系统的技术[J].计算机应用,2009,29(8):2128-2131.

[5] Guo P J，Engler D. CDE:using system call interposition to automatically create portable software packages[C]//Proceedings of the 2011 USENIX conference on USENIX annual technical conference.Berkeley：USENIX association，2011：21-26.

[6] 高明.构建 Citrix XenApp 平台实现移动办公[J].合作经济与科技，2013(5)：120-121.

[7] 张楠.通过桌面云提升高校 IT 应用和管理水平[J].实验技术与管理，2014，31(9)：126-128.

[8] 付婷波.基于 VMware View 图书馆桌面云的设计与实现[J].图书馆理论与实践，2014(6)：96-98.

[9] 钱峥，曹艳艳，赵科科，许皓皓.私有云在市级气象业务平台的实现与应用[J].气象科技，42（4).

[10] 黄荣.基于 Citrix XenApp 的高校校园网应用虚拟化设计及研究[J].计算机时代，2014(3)：17-19.

[11] 张利远，王春丽.基于 Citrix XenApp 虚拟化技术的云教室安全部署研究[J].现代教育技术，2013(7)：95-99.

[12] 杨保华,戴王剑,曹亚仑.Docker 技术入门与实战[M].北京：机械工业出版社,2014.12

[13]Jang Seung-Ju. Design of the Remote Management System in the Windows Operating System[J]. International Journal of Computer Science and Network Security,2011,11(11):38-42.

[14] 崔竞松，何松，郭迟，贺汇林.基于 KVM 虚拟桌面的透明消息通道设计[J].计算机工程，2014，

40(9)：77-81.
[15] James Turnbull.The Docker Book[M]. Amazon Digital Services, Inc.,2014