

基于 CloudStack 和 OpenStack 的 KVM 虚拟机跨平台迁移方法

郑楠¹, 陈立南^{1,2}, 郑礼雄³, 马严^{1,4}

(1. 北京邮电大学 信息网络中心, 北京 100876; 2. 北京信息科技大学 信息管理学院, 北京 100192;
3. 国家计算机网络应急技术处理协调中心, 北京 100029; 4. 移动互联网安全技术国家工程实验室, 北京 100876)

摘要: 在 CloudStack 平台与 OpenStack 平台共存的环境中, 为了使 CloudStack 平台中已创建的 KVM 虚拟机在迁移到 OpenStack 平台后可以被 OpenStack 平台的控制节点正确识别并接管, 提出了一种将 CloudStack 平台中已经存在的虚拟机动态迁移到 OpenStack 平台的方法。通过将传统基于本地存储的 KVM 虚拟机迁移方法与 CloudStack 以及 OpenStack 云计算平台自身特点相结合, 重新对虚拟机迁移相关文件进行组合, 实现了虚拟机跨平台的动态迁移。实验结果表明, 本方法不但可以完成将 KVM 虚拟机成功从 CloudStack 平台迁移到 OpenStack 平台的任务, 而且在时间上与传统方法相比并未产生其他时间成本。

关键词: 虚拟化; 云计算; OpenStack; CloudStack; KVM 虚拟机; 迁移

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2014)Z1-0072-04

Method of KVM virtual machine live migration in cross-platform based on Cloudstack and OpenStack

ZHENG Nan¹, CHEN Li-nan^{1,2}, ZHENG Li-xiong³, MA Yan^{1,4}

(1. Network Information Center, Institute of Network Technology, Beijing University of Posts and Communications, Beijing 100876, China;
2. School of Information Management, Beijing Information Science & Technology University, Beijing 100192, China;
3. CNCERT, Beijing 100029, China; 4. National Engineering Laboratory for Mobile Network Security, Beijing 100876, China)

Abstract: To make KVM virtual machine which have been created in CloudStack platform identified and taken over correctly in CloudStack and OpenStack concomitant environment, propose a method for live migration of KVM virtual machine from CloudStack to OpenStack. By combining traditional KVM virtual machine migration method based on local storage and the characteristics of CloudStack and OpenStack cloud computing platform, regrouping virtual machine migration-related documents, we achieved the dynamic migration of virtual machines across platforms. The experimental results show that the method can not only successfully completed the KVM virtual machine migration task from CloudStack to OpenStack, but also have no more other time cost than the traditional method.

Key words: virtualization; cloud compute; OpenStack; CloudStack; KVM virtual machine; migration

1 引言

随着云计算技术逐渐成熟和不断发展, 各种 IaaS (infrastructure as a service) 云的解决方案不断涌现。其中, OpenStack 和 CloudStack 是应用最广

泛的 2 种搭建 IaaS 云计算平台的开源云技术。但是无论是 OpenStack 还是 CloudStack, 在拥有自身优点的同时也存在一些特定的缺陷。因此, 在企业、科研或教育机构搭建自身私有云或公有云平台的过程中, 单一使用其中一种技术实现整个平台部署

收稿日期: 2014-10-24

基金项目: 国家高技术研究发展计划 (863 计划) (2013AA014702); 北京市教委科研计划基金资助项目 (KM201311232017); 中央高校基本科研业务费基金资助项目 (2014PTB-00-04, 2014ZD03-03); 中国下一代互联网基金资助项目 (CNGI-12-02-027); DNSLAB 基金资助项目

Foundation Items: The National High Technology Research and Development Program of China (863 Program) (2013AA014702); Beijing Municipal Commission of Education Research Project (KM201311232017); Fundamental Research Funds for the Central Universities (2014PTB-00-04, 2014ZD03-03); China Next Generation Internet Project (CNGI-12-02-027); DNSLAB

的方法正在被同时使用以上 2 种 IaaS 云计算平台技术的实现方法所取代。在 OpenStack 和 CloudStack 2 种 IaaS 云计算平台共存的环境中, 一个重要的问题就是怎样在 2 个平台间稳定地实现虚拟机的动态迁移。

虽然目前 OpenStack 和 CloudStack 都已经实现了对 VMware、KVM、XenServer、XCP 等主流虚拟化技术的支持, 但是由于 KVM 技术的强大, 目前在 2 种平台下 KVM 是首选的底层虚拟化技术。文献[1]中提出的传统 KVM 虚拟机迁移包括静态迁移、动态迁移 2 种方式, 动态迁移包括基于共享存储的动态迁移和基于本地存储的动态迁移 2 种实现方法。

静态迁移也被称作常规迁移或离线迁移, 是指在虚拟机关机或暂停的状态下从一台宿主物理主机迁移到另一台宿主物理主机的过程。因为虚拟机的文件系统建立在虚拟机镜像上面, 所以在虚拟机关机的情况下, 只需要简单地迁移虚拟机镜像和相应的配置文件到另外一台物理主机上。如果需要保存虚拟机迁移之前的状态, 在迁移之前将虚拟机暂停, 然后拷贝状态至目的宿主物理主机, 最后在目的宿主物理主机重建虚拟机状态并恢复执行。这种方式的迁移过程需要显式地停止虚拟机的运行。

动态迁移也被称作在线迁移, 是指在保证虚拟机中各种服务正常运行的同时, 将虚拟机系统从一台宿主物理主机移动到另一台宿主物理主机的过程。文献[2]介绍了基于共享存储的动态迁移方法是指在源宿主物理主机和目的宿主物理主机之间采用 SAN (storage area network) 或 NAS (network attached storage) 之类的集中式共享外存设备, 虚拟机运行所需的镜像文件和相关配置文件都存储在中间共享外存设备中。这样, 在迁移过程中只需要将虚拟机内存执行状态迁移到宿主物理主机中即可完成迁移。基于本地存储的虚拟机迁移也被称作块迁移, 是指虚拟机的磁盘镜像文件和相关配置文件存储在源宿主物理主机中, 迁移过程首先要完成将磁盘镜像文件迁移到目的宿主物理主机中, 然后再将虚拟机内存执行状态迁移到宿主物理主机中即可完成迁移。

以上迁移方法虽然都可以完成 KVM 虚拟机的迁移任务, 但是在 OpenStack 和 CloudStack 环境中, 简单使用传统的虚拟机迁移方法无法使云计算平

台中的控制节点识别并对迁移后的虚拟机进行管理。这样将导致用户无法通过管理界面找到被迁移后的虚拟机。虽然, 文献[3]和文献[4]中也提供了一套 OpenStack 和 CloudStack 用来完成自身平台内虚拟机的迁移任务的方法和 API, 但是这种方法具有极强的局限性, 它只能完成 OpenStack 或 CloudStack 云平台内部计算节点之间的虚拟机迁移任务, 无法完成虚拟机的跨平台迁移。

通过将传统基于本地存储的 KVM 虚拟机迁移方法与 CloudStack 以及 OpenStack 云计算平台自身特点相结合, 提出了一种将 CloudStack 平台中的虚拟机动态迁移到 OpenStack 平台的实现方法。在跨平台迁移的实验中证实了该方法可以成功地将 CloudStack 平台中已创建的虚拟机迁移到 OpenStack 平台中并被 OpenStack 平台的控制节点成功接管。同时, 迁移过程的时间损耗基本与传统方法持平, 未增加新的时间成本。

2 KVM 虚拟机迁移

KVM 虚拟机启动时需要读取一个 XML 配置文件, 文件中将指明该虚拟机的 vcpu 个数、内存大小和磁盘大小等内容。其中一个配置选项指明了该虚拟机启动所依赖的磁盘镜像在宿主物理主机中的路径。该磁盘镜像文件在启动 KVM 虚拟机时自动生成, 它保存了在虚拟机启动后用户对虚拟机所做的一切操作。KVM 虚拟机在启动时还依赖另一个磁盘镜像文件, 这个文件是用户导入的操作系统镜像。2 个镜像文件再加上一个配置文件就可以成功地确定一个 KVM 虚拟机。

CloudStack 平台总体上分为 2 个部分, 一个是控制节点, 它负责与用户进行交互并完成对其他计算节点的控制与管理; 另一个是计算节点, 它是真正启动虚拟机的位置。虚拟机的相关启动文件会存储在一个 NFS (network file system) 系统中。控制节点与计算节点都可以由一台或多台物理机组成。当虚拟机启动后, 它是以文件的形式存储在计算节点中的, CloudStack 平台会通过用户选择的相关配置自动生成一个配置文件, 配置文件中的 SourceFile 项指明了保存用户对虚拟机操作的磁盘镜像文件的路径, 打开 SourceFile 文件, 其中的 BackingFile 项指明了用户选择的操作系统磁盘镜像的路径。

与 CloudStack 平台相似, OpenStack 平台也

可以分为控制节点和计算节点两大部分，虚拟机启动后所需要的文件会存储在计算节点中。传统的基于本地存储的虚拟机迁移方法是将源宿主物理主机中的配置文件连同 2 个磁盘镜像文件一起迁移到目的宿主物理主机，同时不停止源宿主物理主机中的虚拟机。当目的宿主物理主机启动完成后关闭并删除源宿主物理主机中的虚拟机。使用这种方法迁移虚拟机仅仅是在 2 个平台的计算节点之间完成了虚拟机相关文件的迁移，迁移后的虚拟机虽然可以正常启动，但是由于全过程没有涉及任何与 OpenStack 平台的控制节点的交互，控制节点并不知道在自己管理的计算节点中多了一台虚拟机，因此用户无法通过 OpenStack 平台的管理界面对迁移到 OpenStack 平台中的虚拟机进行管理和操作，也就是说用户丧失了对虚拟机的控制权。

针对出现的问题，在原方法的基础上进行了改进，在保持传统方法迁移流程的同时，使用 OpenStack 平台的 API 自动生成配置文件并完成操作系统镜像文件的注册。然后使用源虚拟机的相关文件替换 OpenStack 平台自动生成的文件，进而使虚拟机的迁移过程与 OpenStack 平台的相关控制节点产生交互，这样就可以使虚拟机在迁移完成后可以被 OpenStack 平台的控制节点正确地识别并管理。

迁移虚拟机的第 1 步是在 CloudStack 环境中的计算节点内找到所要迁移虚拟机 (VM1) 的 SourceFile 文件 (A 文件)，并通过其找到所对应的 BackingFile 文件 (B 文件)。通过使用虚拟化软件 libvirt 和 qemu，整个过程可以很方便的完成。第 2 步要实现的是利用 B 文件在 OpenStack 环境中创建模板。这个过程可以通过 2 种方法实现。方法一，网络连通的情况下，在 CloudStack 平台的计算节点安装 python-glanceclient 软件，通过 glance image create 命令完成创建。方法二，将 B 文件拷贝到 OpenStack 平台的相关控制节点中，然后使用 glance image-create 命令或者通过用户管理界面完成模板的创建。把通过 B 文件所创建的模板文件叫做 B1。完成以上步骤后，进行第 3 个步骤，通过 nova boot 命令或者通过用户管理界面在 op-enstack 环境中创建一个虚拟机 (VM2)。创建 VM2 的模板文件选择 B1。当虚拟机正常启动后，VM2 也将在 OpenStack 环境 (compute node 节点) 中生成属于它自己的 SourceFile 和 BackingFile，把这 2 个文件分别叫做

C 文件和 D 文件。第 4 步，找到 C 文件的具体位置并使用 A 文件将其替换。A 文件中的 BackingFile 项与 C 文件中的 BackingFile 项所指向的文件是相同的，但是由于 A 文件和 C 文件属于不同的平台，其指向 BackingFile 的路径并不相同，所以，在使用 A 文件替换 C 文件之前必须要做的是提取 C 文件内 BackingFile 项的内容，并用它替换掉 A 文件内的相应位置，然后再完成 A 文件替换 C 文件的过程。此时，CloudStack 平台中的虚拟机 VM1 被成功地迁移到 OpenStack 平台，并可以通过 OpenStack 平台的用户管理界面对其进行访问和控制。

3 实验

为了验证方法的有效性和传统基于本地存储的动态迁移方法在时间损耗上进行对比，选取了 3 种不同大小的磁盘镜像文件，并在 CloudStack 平台下通过每种镜像创建一台虚拟机，然后通过脚本程序分别使用传统基于本地存储的迁移方法和新方法进行实验。本实验使用 OpenStack 的版本是 Havana，CloudStack 版本是 4.3。实验中需要分别搭建 CloudStack 环境和 OpenStack 环境，如表 1 和表 2 所示。其中，OpenStack 环境共包含 3 个节点，CloudStack 环境中共包含 2 个节点。

表 1 OpenStack 环境部署及相关软件

主机名	controller node	network node	compute node
操作系统	centos6.5	centos6.5	centos6.5
安装组件	keystone glance nova management neutron Server ML2 Plug-in horizon	ML2 plug-in layer 2 agent(OVS) layer 3 agent DHCP agent	nova hyper visor KVM ML2 plug-in layer 2 agent(OVS)
其他软件	database(MySQL) message broker (qpid)		
最低配置	单核 CPU 2 GB 内存 5 GB 硬盘	单核 CPU 512 MB 内存 5 GB 硬盘	单核 CPU 2 GB 内存 10 GB 硬盘

表 2 CloudStack 环境部署及相关安装软件

主机名	management	host
操作系统	centos6.5	centos6.5
安装组件	management server	host agent
其他软件	database(MySQL) NFS	KVM
最低配置	单核 CPU 1 GB 内存 20 GB 硬盘	单核 CPU 1 GB 内存 20 GB 硬盘

实验环境部署在 2 个子网中，其中 OpenStack 环境中的 3 台主机处于同一个子网，CloudStack 环境中的 2 台主机处于另一个子网，2 个子网之间可以互相连通如图 1 所示。在 OpenStack 环境中，controller node 与 network node 通过网络管理 compute node，虚拟机实际存储在 compute node 中，用户既可以通过管理界面访问 controller node 从而管理和操作各个虚拟机，也可以通过 API 使用命令行的形式管理和操作各个虚拟机。CloudStack 环境的运行机制与 OpenStack 类似，management 通过网络管理 host，同时也提供了管理界面和 API 2 种方式来方便用户对存储在 host 中的虚拟机进行一系列的管理与操作。

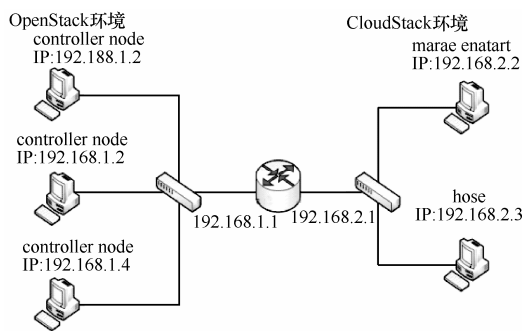


图 1 实验网络拓扑环境

首先使用传统基于本地存储的虚拟机迁移方法进行实验，然后再使用新方法进行实验。整个实验过程中网络拓扑环境和软件环境不发生变化。实验结果如表 3 和表 4 所示，2 种方法都可以将虚拟机全部迁移到新的环境中，但是相对于传统基于本地存储的迁移方法，通过新方法迁移到 OpenStack 平台的虚拟机全部可以通过 OpenStack 平台的管理界面被访问到。在时间损耗方面，新方法与传统方法基本持平，并未增加额外的时间损耗。

表 3 基于本地存储迁移方法的实验结果

镜像文件名	文件大小 /MB	可否被管理界面访问	迁移时间/s
cirros-x86_64 qcow2	9	否	4
fedora-x86_64 qcow2	200	否	200
ubuntu_mysql qcow2	450	否	325

表 4 新迁移方法实验结果

镜像文件名	文件大小 /MB	可被管理界面访问	迁移时间/s
cirros-x86_64 qcow2	9	是	4
fedora-x86_64 qcow2	200	是	202
ubuntu_mysql qcow2	450	是	325

4 结束语

通过将传统基于本地存储的 KVM 虚拟机动态迁移方法与 CloudStack 平台以及 OpenStack 平台自身特性相结合，实现了 KVM 虚拟机跨 CloudStack 平台迁移到 OpenStack 平台的一种方法。与传统方法相比，新方法解决了传统方法跨平台迁移后虚拟机无法被 OpenStack 平台的控制节点识别的问题。进一步的研究工作主要从以下 2 个方面着手进行：一是在保证迁移成功的前提下提高效率，减小虚拟机服务的切换时间；二是进一步优化方法，使迁移过程可以结合平台自身的资源调度策略，从而避免在单个计算节点中启动过多的虚拟机而影响宿主机物理主机的运行速度。

参考文献：

- [1] CHRISTOPHER C, KEIR F, STEVEN H, et al. Live Migration of Virtual Machines[C]. <http://wenku.baidu.com/view/2208efc42cc58bd63186bd66.html>, 2005.
- [2] YAMUNADEVI L, ARUNA P, SUDHA DEVI D. Security in virtual machine live migration for KVM[A]. Process Automation, Control and Computing (PACC)[C]. 2011.7: 20-22.
- [3] OpenStack admin user guide[EB/OL]. <http://docs.OpenStack.org/user-guide-admin/content/>, 2014.
- [4] Apache cloudStack administration documentation[EB/OL]. <http://docs.cloudstack.apache.org>, 2014.

作者简介：



郑楠 (1987-)，男，河北张家口人，北京邮电大学硕士生，主要研究方向为云计算、下一代互联网关键技术等。

陈立南 (1972-)，女，黑龙江勃利人，北京邮电大学博士生，北京信息科技大学副教授，主要研究方向为新一代互联网关键技术等。

郑礼雄 [通信作者] (1982-)，男，江西玉山人，国家计算机网络应急技术处理协调中心工程师，主要研究方向为网络安全威胁监测与防御技术。E-mail: zlx@cert.org.cn。

马严 (1955-)，男，北京人，北京邮电大学教授、博士生导师，主要研究方向为基于 TCP/IP 网络的网络管理技术、网络安全技术、移动 IP 技术、IPv6 技术及其应用。