

容器技术及其在OpenStack云平台的应用

近两年来，容器技术作为一项轻量级虚拟化解决方案成为云计算领域热门的研究方向，其与OpenStack等云平台的结合受到普遍关注，业界各大巨头公司均表示浓厚兴趣并参与其社区贡献中。

沈雪晴 朱洪斌 邢宇恒/国家电网公司信息通信分公司

0 引言

2013年Docker容器开源项目的诞生带领容器技术成为云计算领域热门的研究方向，其与OpenStack等云平台的结合受到普遍关注，业界各大巨头公司均表示浓厚兴趣并参与其社区贡献中，国内阿里云ECS、国外亚马逊以及谷歌等均发布容器服务，支持在云平台实例集群上发布、管理以及扩展Docker容器。OpenStack作为业界领先的开源云平台项目，由于其具有开源性、灵活性、可扩展性、强大的功能及社区支持等诸多特性，成为当今云平台领域的热门研究内容，OpenStack目前正在推进其与Docker容器技术的融合，以其自身强大的架构、存储和网络等各方面支持Docker容器。本文将对容器技术中的佼佼者Docker开源项目的功能架构、关键技术和生态系统进行介绍并比较其与传统虚拟化方式的区别，同时对其与当前各大云计算平台的结合进行研究，尤其是与业界领先的开源云平台OpenStack的结合，思考其未来发展趋势及对IT发展各个领域的启示。

国家自然科学基金（00000000）：国家高技术研究发展计划（863计划）（2008AA000000）。



沈雪晴/助理工程师

关键词/Keywords

云计算 ·
OpenStack ·
虚拟化 ·
容器 ·
Docker ·

1 Docker容器技术

2013年，Docker 版本0.1发布：Docker最初是平台即服务（Platform-as-a-Service，PaaS）提供商dotCloud开源的一个基于Linux容器（Linux Container，LXC）的高级容器引擎，源代码托管在GitHub上，基于Go语言并遵从Apache 2.0协议开源。Docker目前正处于高速发展中，当前最新版本为2015年11月初发布的1.9版本，该版本在集群管理、网络互连和存储等方面均进行了重大升级，为扩展分布式应用产品提供了基础。

1.1 基本概念

Docker包括三个基本概念：镜像（Image）、容器（Container）和仓库（Repository）。

1）镜像。Docker镜像就是一个模板，例如：一个镜像可以包含一个完整的ubuntu操作系统环境，里面仅安装了Apache或用户需要的其它应用程序。镜像可以用来创建Docker容器。

2）容器。Docker利用容器来运行应用。容器是从镜像创建的运行实例。每个容器都是相互隔离、保证安全的平台。

3）仓库。仓库是集中存放镜像文件的场所。每个仓库包含多个镜像，每个镜像有不同的标签（tag）。仓库分为公开仓库（Public）和私有仓库（Private）两种形式。最大的公开仓库是Docker Hub，存放了数量庞大的镜像供用户下载^[1]。

1.2 基础架构

Docker 采用了 C/S架构，如图1所示，包括客户端和服务端。Docker后台作为服务端接受来自客户的请求，并处理这些请求（创建、运行、分发容器）。客

户端和服务端既可以运行在一个机器上，也可通过Socket 或者RESTful API 来进行通信^[2]。

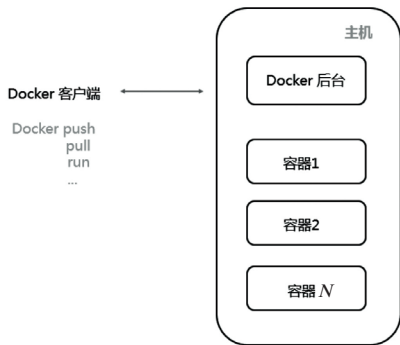


图1 Docker C/S架构

Docker服务器一般在宿主主机后台运行，等待接收来自客户端的消息。Docker客户端则为用户提供一系列可执行命令，用户用这些命令实现跟Docker后台交互。Dockerfile是一个镜像的表示，包含了用户指定的软件依赖、使用卷等，可以通过Dockerfile来描述构建镜像的步骤。当执行Docker build命令时，按照Dockerfile描述，生成镜像；Docker run命令基于指定镜像生成容器并启动它，容器可以执行stop、start和restart等操作；当执行commit命令时，将提交当前容器为新的镜像；push和pull命令用户和Docker仓库的交互，push推送指定镜像或者库镜像至docker源服务器，pull从Docker源服务器拉取制定镜像。Save和load命令则是将镜像保存为本地tar包，或者从tar包中解压下载。

1.3 底层技术支持

Docker目前利用的是Linux内核本身支持的容器方式实现资源和环境隔离。简单的说，docker利用名字空间（Namespace）实现系统环境的隔离；利用控制组（Cgroup）实现资源限制；利用镜像实现根目录环境的隔离。目前Docker已经将底层实现都抽象化到Libcontainer的接口。底层容器的实现方式变成了一种可变的方案，名字控件技术抑或是使用Systemd等其他方案，只要实现了Libcontainer定义的一组接口，Docker都可以运行。这也为Docker实现全面的跨平台带来了可能。

1.4 与传统虚拟化方式区别

容器和普通的虚拟机相比，最大的区别是它并不包含操作系统内核。无论是KVM还是Xen，在容器看来都在浪费资源，因为用户需要的是高效

运行环境而非操作系统。

图2显示了Docker和传统虚拟化方式的实现框架，Docker并没有和虚拟机一样利用一个完全独立的Guest OS实现环境隔离，它利用的是目前Linux内核本身支持的容器方式实现资源和环境隔离。容器新建只需要几秒钟时间。此外，由于Docker不需要Hypervisor实现硬件资源虚拟化，运行在Docker容器上的程序直接使用的都是实际物理机的硬件资源。因此在CPU、内存利用率上Docker均有优势^[3]，下表展示了Docker和传统虚拟机在性能上的具体差异。

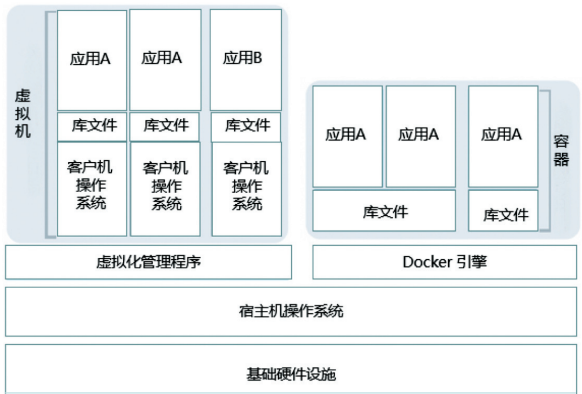


图2 容器及传统虚拟化技术对比

表 Docker容器和传统虚拟机方式差异

	容器	虚拟机
操作系统	与宿主主机共享操作系统	宿主主机操作系统上运行虚拟机系统
存储大小	镜像效，便于存储于传输	镜像庞大（vmdk、vdi等）
运行性能	几乎无额外性能损失	操作系统额外的CPU、内存消耗
硬件亲和性	面向软件开发者	面向硬件运维者
部署速度	快速，秒级	较慢，10 s以上

2 Docker与OpenStack融合

图3显示了OpenStack中和Docker有关系项目。

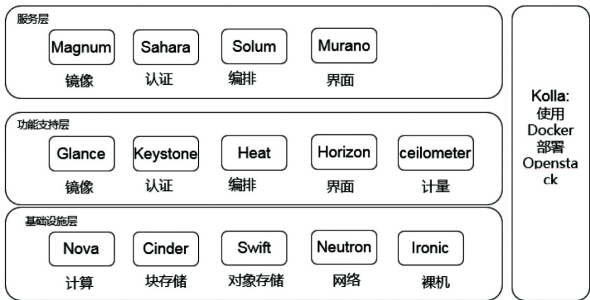


图3 OpenStack中和Docker有关系项目

Nova计算、Heat编排以及Magnum容器即服务使用Docker容器对外提供服务，而Sahara大数据、Solum应用开发、Muarno应用目录和Kolla部署OpenStack均是使用Docker为项目自身进行服务。

下面主要介绍当前应用较广的Docker与Nova集成以及OpenStack未来大力推广的容器项目Magnum，不管是将Docker与Nova集成，还是将Docker与Magnum集成，其本质上都是将Docker作

2.1 Docker与Nova集成

OpenStack支持Docker与其计算项目Nova相结合，将Docker作为类似KVM及Xen一样的虚拟机资源使用；Nova容器驱动实现比较简单，仅需要实现Nova计算中的一些对虚拟机操作的常用接口。目前主要支持创建、启动、停止、暂停和回复等虚拟机的基本操作。Nova容器驱动像其他的计算驱动一样使用OpenStack中的所有服务，包括用Heat来做应用部署、服务发现、扩容缩容等，和Neutron集成来管理Docker网络等。也支持多租户，为不同的租户设置不同的Quota，做资源隔离。

其缺点也很明显，因为Docker和虚拟机差别较大，Docker与Nova集成后，无法实现Docker的高级特性，例如容器关联、端口映射、以及Docker诸多网络模式，包括主机模式、容器模式等^[4]。

2.2 Magnum容器服务

OpenStack社区于2014年11月成立了专门针对Docker和OpenStack集成的项目Magnum，用来提供容器服务。类似于Nova是一套支持不同虚拟机管理程序（KVM、VMWare等虚拟机平台）的框架，Magnum则是支持不同容器机制的框架，它同时可以和多个容器集群管理系统集成。Magnum现在并没有调度模块，对容器的调度，完全是通过容器集群管理系统本身的调度器来工作的^[5]，Magnum只是负责将用户创建容器的请求进行转发，转发到对应的容器集群管理系统，最终请求由具体的容器集群管理系统去调度。Magnum现在对Docker网络的管理，也是通过上层的容器集群管理系统自己去管理，目前Magnum主要是对不同的容器集群管理系统的API做了一些封装，后续将逐步促进Magnum组件与OpenStack其他组件以及整个生态系统的融合。其最终目标是期望和Nova一样，能够让用户不用关心后台的Docker集

群管理工具到底是什么，只管提请求，最终通过Magnum获得自己需要的容器。

3 未来发展趋势及启示

由于Docker可以运行在任何有合理数据内核的Linux虚拟机上，因此它可以运行在很多IaaS提供的云上。目前包括阿里云、亚马逊云平台和谷歌云平台等对于Docker的支持主要是将容器作为服务使用，Docker负责向上面向应用，平台负责向下面向硬件，Docker只专注于解决应用部署、扩容等问题，平台负责管理调度网络、存储等硬件资源。这种硬件、云平台 and Docker容器的三层架构可以满足云信息化时代的所有应用场景。

1) 微服务。微服务架构（Microservices Architecture）是一种架构风格（Architectural Style）和设计模式，可以基于IaaS，也可以基于PaaS。微服务架构提倡将应用分割成一系列细小的服务，每个服务专注于单一业务功能，运行于独立的进程中，服务之间边界清晰，采用轻量级通信机制（如HTTP/REST）相互沟通、配合来实现完整的应用，满足业务和用户的需求。微服务模式下系统的每个组件均可以独立部署，在生产环境的发布阶段，即便局部发生故障也不必停止其他部分的系统工作。

微服务架构带来了诸多优势，但必须承认，构建、部署及维护分布式的微服务系统并不容易。容器Docker的出现将所有应用都标准化为可管理、可测试且易迁移的镜像/容器，其所提供的轻量级、面向应用的虚拟化运行环境为微服务提供了理想的载体。同样，基于容器技术的云服务将极大的简化容器化微服务创建、集成、部署和运维的整个流程，从而推动微服务在云端的大规模实践。

2) 持续集成/持续交付(CI/CD)。尽管Puppet、Chef、Salt以及其他先行者已经在开发和运维（Development Operations, DevOps）的探索中作出了相当的贡献，但这些工具的主要活动舞台仍然集中在运营团队而非开发者群体当中。Docker是第一款能够在开发人员群体中获得与运营工程师同等认可效果的DevOps工具。开发人员可以在容器机制内部工作，而运营工程师则能够在容器外部以并行方向处理自己的任务。开发团队采纳Docker的同时，

相当于在软件开发生命周期当中添加了新的敏捷性层。容器技术最大的突破就在于其一致性。基于Docker的应用程序能够在笔记本电脑与生产环境下获得同样的运行效果。由于Docker将与应用程序相关的所有状态都封闭了起来，因此不用担心底层操作系统的架构差异会给应用程序的运行依赖性带来影响或是产生新的漏洞。

4 结束语

Docker是容器技术中的佼佼者，其使用类似集装箱的思想将各种应用程序和他们所依赖的运行环境打包成标准的镜像，进而发布到不同的平台上运行，极大地方便了应用的开发及部署。同时，Docker作为一个容器级虚拟化的解决方案，其比传统虚拟化方式更加轻量，具有更高的性能、灵活性以及可扩展性。

未来云计算将逐步发展为Docker负责向上面向应用，平台负责向下面向硬件，Docker只专注于解决应用部署、扩容等问题，平台负责管理调度网络、存储等硬件资源。这种硬件、云平台和Docker容器的三层架构可以满足云信息化时代的所有应用场景。Docker正在迅速改变云计算领域

的运作规则，并彻底颠覆云技术的发展前景。

参考文献

[1] 戴王剑, 杨保华, 曹亚仑. Docker技术入门与实践[M]. 北京: 机械工业出版社, 2015.

[2] Docker介绍以及其相关术语、底层原理和技术[EB/OL]. <http://www.open-open.com/lib/view/open1413190983463.html>.

[3] Docker与虚拟机性能比较[EB/OL]. <http://blog.csdn.net/cbl709/article/details/43955687>.

[4] 容器不会取代OpenStack, 但二者如何深度整合?[EB/OL].<http://www.csdn.net/article/2015-05-20/2824734>.

[5] DockOne技术分享: OpenStack Magnum社区及项目介绍[EB/OL]. http://dockone.io/article/445utm_source=tuicool.

作者简介

沈雪晴（1990-），女，硕士，主要研究方向为IT运维、云计算和Web服务接纳控制研究。

（收稿日期：2015-12-10）
（编辑：崔文静）

（上接第540页）

2 结束语

目前，天津电力云终端正处在部署阶段，相较于传统的人手一台PC机的老式办公手段，云终端具有以下优势：

- 1) 设计灵巧，云终端部署完成后，每个办公桌都将去掉传统的主机机箱，取而代之的是小小的云终端客户机，极大节约空间。
 - 2) 使用灵活，在每个部署有云终端的工位，登陆自己的账号后，都将展现自己设置的桌面，破除了空间的局限性。
 - 3) 节约能源，统一部署云终端后，每个云终端的功耗相比于传统PC主机有很大的低能耗优势。
 - 4) 维护方便，由于集中部署了云终端系统，方便了信息运维人员对于日常PC机的维护。
- 天津电力公司在云终端部署完成后，将大大

提升信息化办公的效率，降低PC机损耗率，降低日常运维成本，落实公司“集约化”管理要求，提高对员工终端的管控力度。

参考文献

[1] 钟志琛, 邹海彬. 云终端提升电力企业桌面中断安全技术探讨[J]. 电力信息与通信技术, 2013(9).

[2] 韩向春, 潘勋, 陈晶, 等. 计算网格中动态负载均衡的分布调度模式[J]. 计算机工程与设计, 2007(12).

作者简介

于晓冬（1989-），男，本科，助理工程师，从事电网信息通信调度工作。

（收稿日期：2015-12-10）
（编辑：崔文静）