



基于开源 PaaS 技术的互联网业务平台自动部署方案

何震苇,严丽云,李慧云,张 凌,陆 钢

(中国电信股份有限公司广州研究院 广州 510630)

摘 要: 从互联网业务平台在云环境下的部署架构和部署需求入手,分析了 Docker 和 Cloudify 等新兴开源 PaaS 平台的集群部署解决方案,在此基础上设计了支持全堆栈、全流程的互联网应用集群自动部署方案,以简化复杂互联网业务平台的部署过程。

关键词: 自动部署;PaaS;互联网业务平台;TOSCA;容器

doi: 10.11959/j.issn.1000-0801.2015259

Automatic Deployment Solution for Internet Business Platform Based on Open-Source PaaS

He Zhenwei, Yan Liyun, Li Huiyun, Zhang Ling, Lu Gang

(Guangzhou Research Institute of China Telecom Co., Ltd., Guangzhou 510630, China)

Abstract: Starting with the deployment architecture and deployment requirements of internet business platform in cloud environments, the deployment solution of the rising up open-source PaaS platform was analyzed, such as Docker and Cloudify. An open-source based automatic deployment solution was proposed for internet business platform with full-stack and full-flow support, which simplify the deployment of complex internet business platform.

Key words: automatic deployment, PaaS, internet business platform, TOSCA, container

1 引言

高速发展的互联网业务平台面临用户需求频繁变化和用户规模快速增长的问题,互联网业务平台需要具备快速部署、可靠运行和弹性扩展的能力。随着互联网业务平台的发展趋于技术多样化、架构复杂化和集群规模化,其部署管理也变得越来越困难。云计算 IaaS 技术实现了业务平台硬件资源的池化共享,简化了业务平台在硬件层面上的部署过程。云计算 PaaS 技术则实现了中间件数据库

等平台软件的池化共享,在一定程度上也简化了业务平台软件层面的部署过程。然而传统纯 IaaS 和纯 PaaS 的应用部署方式都存在一定的局限性,纯 IaaS 方式基于虚拟机镜像模板,缺乏对基础软件和应用软件部署层面的支持,需要较多的软件手工安装配置操作,部署效率较低,而纯 PaaS 方式则无法自主选择集群的架构和依赖的基础软件,部署的灵活性较差。

Docker 和 Cloudify 等新兴 PaaS 平台的出现,为解决上述问题提供了构建简单、灵活、高效、全堆栈、全流程的互

收稿日期:2015-06-16;修回日期:2015-10-09

论文引用格式:何震苇,严丽云,李慧云等. 基于开源 PaaS 技术的互联网业务平台自动部署方案. 电信科学, 2015259

He Z W, Yan L Y, Li H Y, et al. Automatic deployment solution for internet business platform based on open-source PaaS. Telecommunications Science, 2015259



联网业务平台部署方案的可能。

2 互联网云业务平台部署架构与需求分析

互联网云业务平台通常由若干个服务器集群构成,每个服务器集群又由若干个服务器节点构成,每个服务器节点又由硬件资源和包含若干层软件的软件栈构成,在云环境中部署的业务平台硬件依赖于底层云资源池的计算、存储和网络资源。互联网云业务平台的典型部署架构如图1所示。

互联网云业务平台的部署需要处理3种依赖关系:软件依赖关系、拓扑依赖关系和资源依赖关系。

(1) 软件依赖关系

业务平台节点的软件通常由多个层次的软件堆栈构成,上层软件的安装依赖于下层软件的安装。整个软件堆栈从下到上一般可分为4层:操作系统、平台软件、技术框架和应用程序。操作系统通常是类Linux操作系统,如Ubuntu、CentOS;平台软件通常是数据库、Web服务器等拥

有独立的进程空间和服务端口的服务器软件,如Tomcat、MySQL等;技术框架是依附在平台软件上,提供特定语言相关的软件功能的框架或类库,如SSH(Struts+Spring+Hibernate)、Log4J等;应用程序是指实现业务逻辑、提供业务相关功能的程序或软件分组。

在业务平台的部署环境中,同一个集群的节点、不同集群的节点甚至不同业务平台的节点可能采用相同或类似(至少有两层相同)的软件栈,例如图1中负载均衡集群和Web服务集群中的负载均衡器就采用完全相同的Nginx软件栈。而另一方面,不同的互联网业务平台甚至同一个平台的类似集群有可能使用不同的技术框架,例如图1的Web服务集群A使用Tomcat+SSH框架实现Web页面服务,而Web服务集群B则使用NodeJS+DozerJS框架实现Web API服务。

(2) 拓扑依赖关系

大型互联网业务平台往往涉及多个集群,每个集群又由若干个节点或子集群构成,节点与节点之间通过服务端

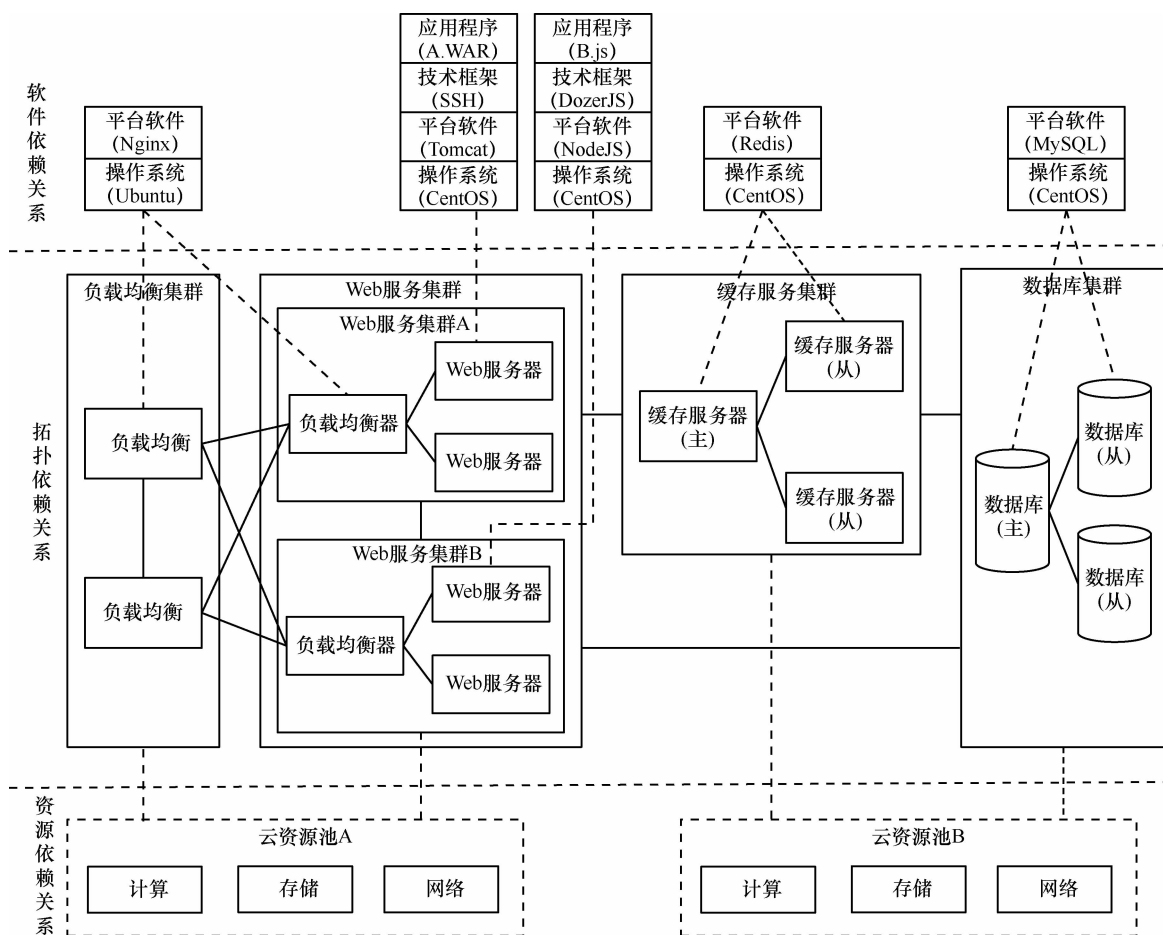


图1 典型互联网云业务平台部署架构

口通信,业务平台的部署需要建立业务平台的拓扑结构,解决节点与节点、节点与集群以及集群与集群之间的关系。节点与节点的关系一般是通信关系,源节点通过服务端口与目标节点建立通信连接,平台部署时需要将目标节点的服务端口(如 IP、端口、URL 等)配置到源节点上,并保证源节点与目标节点的 3 层链路可通达。节点与集群的关系通常是从属关系,包括网络从属和节点从属,网络从属是指集群的节点与集群属于同一子网,节点从属是指集群的节点由集群的某个主节点管理或控制,集群的部署需要根据从属关系对网络和集群主节点进行相应的配置,同时通过从属关系可对整个集群的节点进行批量操作,如批量配置、批量重启。集群与集群的关系可以是子从属关系或者通信关系,集群从属关系的建立需要将子集群的子网加入父集群的子网中,将子集群的主节点服务端口加入父集群主节点配置中,集群的通信关系需要解决不同集群子网之间的路由转发问题。

(3) 资源依赖关系

互联网云业务平台通常基于共享的云资源池构建,业务平台的部署需要估算业务平台的资源需求,根据资源需求从云资源池中申请恰当规格和额度的虚拟机、虚拟存储和虚拟网络资源,将业务平台的节点和集群与底层的云资源绑定。在大规模、混合云的部署场景中,业务平台的各个集群可能会由不同的云资源池甚至不同的云资源平台提供资源,业务平台部署不仅要适配各种类型的云资源接口,还需要屏蔽不同云资源池的资源接口差异性。

互联网业务平台的自动部署就是在平台的部署过程中尽可能自动地解决上述的软件依赖关系、拓扑依赖关系和资源依赖关系,由部署平台自动完成业务平台的各种资源、子集群、节点和软件的分配、安装和配置,并将开发团队提交的应用包加载到相应的应用节点上,最终让整个业务集群准确无误地运行。

互联网业务平台的自动化部署的关键需求如下。

(1) 自动处理业务节点的软件依赖关系

能够按照软件堆栈的层次结构自动完成业务节点整个软件栈的各层软件的安装、配置和启动,可复用具有相同或相似层次软件栈的安装步骤。

(2) 自动处理业务平台的拓扑依赖关系

能够方便地建立业务平台的拓扑模型,描述平台的集群、子集群、节点之间的从属关系、通信关系和依赖关系,根据从属关系和依赖关系递归完成整个集群的部署,根据

通信关系自动完成节点和网络的相关参数配置。可复用具有相同结构的集群拓扑和部署流程。

(3) 支持主流的云资源平台

支持 OpenStack、CloudStack、vSphere 等业界主流的云资源平台,提供一致的计算、存储和网络资源访问接口。

(4) 支持任意语言/框架/架构

支持任意语言、任意技术框架和任意拓扑架构的互联网云业务平台自动部署。

(5) 提升业务平台的部署密度和部署效率

在相同规格的服务器上能够部署更多的节点,压缩节点的软件栈以提升单个节点的部署效率,增加平台节点部署的并行度,以提升整个集群的部署效率。

3 新兴的开源 PaaS 部署平台

在诸多开源 PaaS 平台中,Docker 和 Cloudify 是两款热门的与环境、语言无关的 PaaS 平台,可用于部署任意语言开发的互联网应用,前者侧重于软件层面的镜像化部署,后者侧重于全堆栈的标准化部署。

3.1 容器级部署平台 Docker

Docker 是一个开源的容器引擎,遵从 Apache 2.0 开源协议。Docker 像集装箱一样将应用及其相关依赖进行整体打包,部署到任何支持 Docker 的环境中运行,让应用的开发、部署和运维都变得简单。Docker 自 2013 年开源后不久就得到了 Google、微软、IBM、RedHat 等业界巨头公司的关注和支持,在 2014 年已成为关注程度排名第二的开源软件,仅次于 OpenStack。Docker 将各种环境依赖和应用程序一起打包成标准的镜像文件,利用轻量级的容器虚拟化技术进行资源隔离和应用运行,多个容器共享同一个操作系统,从而可以快速进行应用的部署、启动、销毁和移植,并降低每个应用实例的资源开销。

Docker 平台可以解决互联网业务平台部署的软件依赖关系和部署效率问题。

(1) 简化业务节点的软件依赖关系

Docker 可以将业务节点软件栈上的一层或多层软件打包成镜像,记录镜像之间的层次依赖关系,多个镜像可以像积木一样层层叠加,部署软件时只需要指定顶层镜像,Docker 就能根据镜像的依赖关系从下往上依次安装镜像,从而完成整个节点软件栈的安装部署。

(2) 标准化节点软件的管理和部署接口

由于业务模块及其依赖关系被打包成标准化的镜像,



可以统一各类软件的打包、存储、安装和运行接口,降低节点软件自动化部署难度。

(3)提升业务节点的部署速度

由于不需要部署 Guest 操作系统,Docker 镜像大小通常为 MB 级,无论是镜像的打包、传输,还是安装的速度都比传统的虚拟机镜像方式要快得多。同时,Docker 将业务节点的软件栈标准化,将其进行整体打包后存储在 Docker 镜像仓库中,部署时只需要从 Docker 镜像仓库中获取需要的镜像进行移植即可。因此,Docker 将传统的应用部署流程“安装—配置—运行”转变为“复制—运行”,从而大大提升了业务节点的部署速度。

3.2 基于标准的全堆栈部署平台 Cloudify

Cloudify 是一个支持 TOSCA (topology and orchestration specification for cloud application) 标准的云编排和部署平台,同样遵循 Apache 2.0 开源协议。TOSCA 提供标准化的云应用编排模型,由标准化组织 OASIS 制定,可用于构建云环境下的各类复杂应用集群的全生命周期管理模型,通过结构化的描述语言定义应用节点能力、应用软件堆栈和应用节点关系,通过基于工作流的管理计划支撑云应用的部署、调度、升级自动化。

Cloudify 是第一个支持 TOSCA 1.1 规范的开源部署平台,提供基于 TOSCA 模型的应用集群部署全流程支持,包括应用集群编排、集群部署、集群监控、弹性调度和应用升级等环节。同时,Cloudify 也是一个全面支持硬件、网络 and 软件部署的全堆栈部署平台,不仅集成了 OpenStack、CloudStack 等 IaaS 层的云资源管理平台,实现硬件资源、NFV 网络的分配与配置,还集成了 Puppet、Chef 等软件配置平台,可实现业务平台软件的安装和配置。

Cloudify 平台可以解决互联网业务平台集群拓扑依赖关系和云资源依赖关系问题。

(1)复杂业务集群的编排与模板化

采用 TOSCA 的 YAML 模型对业务集群的架构、节点、关系以及所依赖的硬件、软件资源和配置脚本进行形式化描述,形成应用集群模板,将复杂的业务集群变成结构化、可视化、可复用的模型。

(2)无缝对接主流云资源平台

Cloudify 从 3.0 版本采用与 OpenStack 一样的 Python 语言重写,能够与 OpenStack 进行代码级的无缝集成,可实现计算、存储、网络资源的分配、设置和调度,同时 Cloudify 也提供了官方的插件与 CloudStack、vSphere、AWS、MS

Azure 等主流 IaaS 平台对接。

(3)平台部署全流程自动化

由 Cloudify 的工作流引擎驱动 TOSCA 模型中的部署流程,以串行或并行的方式依次调用各个节点的管理操作 API,执行节点的安装、配置和启动的任务,从而完成整个集群的自动化部署。

4 互联网应用集群自动化部署平台设计

互联网应用集群自动化部署平台通过 Docker 镜像技术打包应用节点的软件依赖关系,通过 TOSCA 模型定义应用集群的拓扑依赖关系,通过 Cloudify 适配底层云资源,并驱动整个部署流程的执行。集群自动部署平台需遵循以下设计原则。

- 高内聚、松耦合:实现相同职责的业务逻辑聚合在同一个模块中,模块与模块之间通过接口或消息机制解耦。
- 高可用、高容错:部署平台的核心部件采用集群架构部署,以避免单点故障,同时部署的中间状态应持久化保存,即使系统宕机也能快速恢复部署流程。
- 轻量级、易扩展:部署平台由轻量级的内核和可扩展的插件构成,平台内核仅提供核心的部署模型管理和部署流程驱动功能,保持平台架构与核心部件相对稳定,云资源适配、资源调度、资源监控等功能通过插件机制扩展。
- 开源化、低成本:借鉴互联网业界的最佳技术实践,尽量采用成熟度高、活跃度高、应用广泛开源 PaaS 平台框架构建部署平台,避免重复“造车轮”,以降低平台的构建成本,提高平台构建效率。

互联网云应用集群自动部署平台基于开源云计算平台 Docker、Cloudify 和 OpenStack 构建,整个互联网应用集群部署平台由门户、PaaS 管理器、IaaS 管理器、镜像仓库和应用集群 5 部分构成,体系架构如图 2 所示。其中镜像仓库存储了应用集群所需的操作系统、基础软件、技术框架和应用软件等镜像,IaaS 管理器以 API 的形式提供应用集群计算、存储和网络资源的申请、配置、调度和销毁服务,PaaS 管理器提供 TOSCA 部署模型解析、部署流程执行、资源监控和弹性调度等自动部署的核心功能,应用集群节点通过预装的虚拟机监视器(VMM)和 Docker 引擎实现应用软件栈的加载和运行。PaaS 管理器通过调度流程引擎和 TOSCA 容器,执行应用集群部署流程,通过管理代理调用

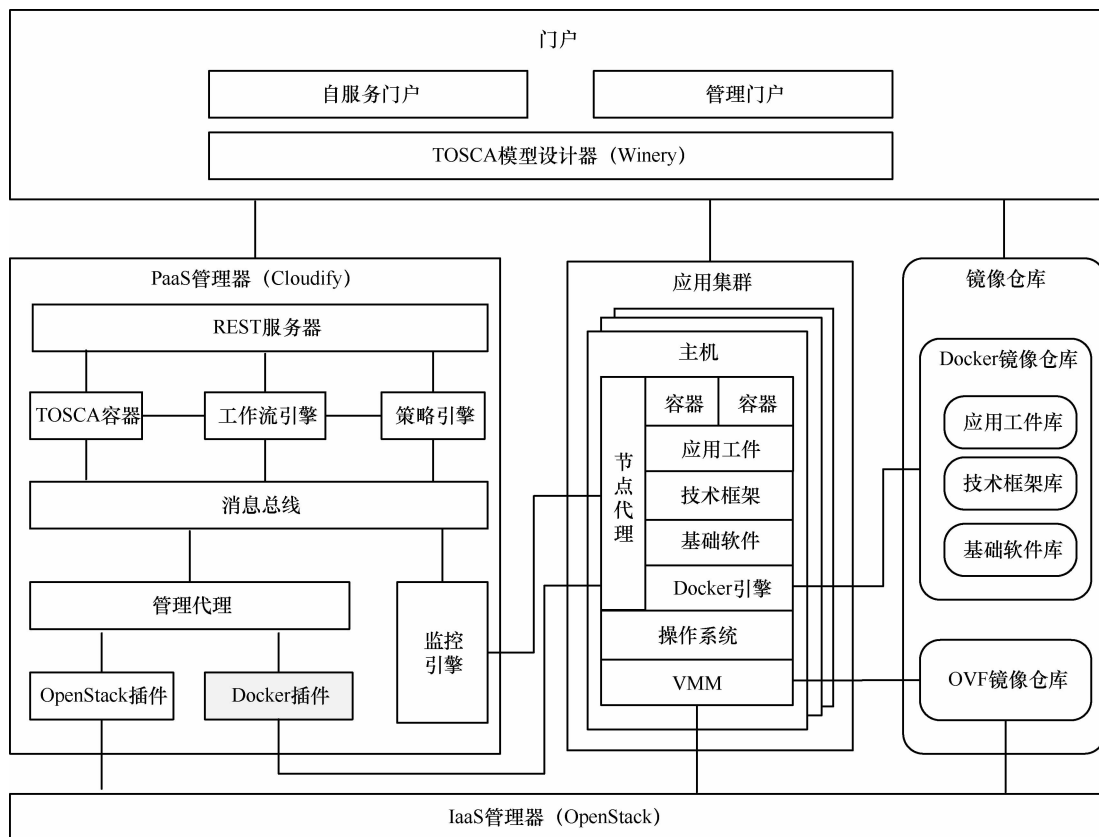


图2 互联网业务自动化部署平台总体架构

OpenStack 和 Docker 引擎的管理接口,完成整个应用集群的部署。

4.1 镜像仓库

镜像仓库负责存储 OVF 系统镜像、基础软件镜像、技术框架镜像库和应用工件镜像,整个镜像仓库由 OVF 镜像仓库和 Docker 镜像仓库构成。OVF 系统镜像仓库存放应用节点的系统级镜像,其内容包括操作系统、Docker 引擎和 Cloudify 节点代理,该镜像仓库由 OpenStack 的镜像管理模块 Glance 实现;Docker 镜像仓库存放基础软件、技术框架和应用工件等轻量级的 Docker 层次化镜像,基于 Docker registry 模块实现。

4.2 PaaS 管理器

PaaS 管理器是整个部署平台的核心模块,基于最新的 Cloudify 3.1 稳定版构建,实现了基于 TOSCA 模型的应用集群部署、监控和调度服务。整个 PaaS 管理器由 REST 服务器、TOSCA 容器、工作流引擎、策略引擎、消息总线、管理代理和管理插件构成。REST 服务器将 PaaS 管理器的所有服务封装成 REST API,向门户提供轻量级、标准化的服务调用接口。TOSCA 容器负责 TOSCA 模型的解析、存储和

加载,为流程引擎和上层应用提供应用集群的静态元数据和运行时状态数据。工作流引擎负责执行 TOSCA 模型定义的各种管理流程,包括集群部署、弹性调度和故障恢复等。策略引擎根据平台采集的监控指标进行操作决策,并通过工作流引擎或管理代理执行具体的管理操作。管理代理封装了各类软硬件资源的管理、配置、激活和调度接口,平台的管理能力可通过管理插件扩展,通过 OpenStack 插件无缝对接 OpenStack 平台,实现计算、存储、网络资源的分配和调度,通过 Docker 插件对接 Docker 引擎,实现对 Docker 容器和镜像的管理。需要指出的是,Cloudify 3.1 尚未提供官方的 Docker 管理插件,需要基于 Cloudify 的插件机制开发 Docker 插件才能对接 Docker 平台,实现 Docker 镜像安装和应用容器运行。

4.3 Docker 引擎

Docker 引擎是驱动 Docker 容器加载运行应用软件栈的核心模块,它驻留在应用节点上,负责 Docker 镜像的上传下载和 Docker 容器的启动销毁。Docker 引擎是应用节点上的一个守护服务进程,该守护进程在后台启动了一个服务进程,接收 Docker 客户端(即 Docker 管理插件)发送的



请求,根据客户端的请求类型,通过路由与分发调度机制执行相应的任务,进行镜像和容器的处理。

4.4 部署平台核心部件的协作关系

通过一个简单的示例描述部署平台在云应用集群部署过程中核心部件的协作关系。假设要部署的应用集群由一个负载均衡器、两个 Web 服务集群和一个数据库服务器构成,其中每个 Web 服务器集群由 2~4 个 Web 服务器构成。如图 3 所示,应用集群的拓扑模型采用 TOSCA 规范描述,每个方框表示一个逻辑节点(可以是集群或服务节点),每个箭头表示节点的依赖关系或通信关系,每个节点由名字、属性和操作构成,部分节点列出了节点的镜像、节点的最小实例数、最大实例数,例如“(服务 A,2:4)”表示该节点依赖的镜像名称为服务 A,最小实例数为 2,最大实例数为 4。应用集群拓扑模型由 TOSCA 容器加载,当拓扑模型加载到内存之后,模型节点的属性和节点的操作就可以通过 REST API 进行访问。应用集群的部署流程由基于

BPEL (business process execution language) 的流程模型描述,流程模型由任务节点、控制节点和有向连接构成,任务节点通过部署模型的 API、其他任务节点或者子流程完成若干部署步骤,控制节点实现部署过程的分支和循环控制,有向连接定义了任务节点的执行次序。应用集群部署流程由流程引擎负责加载和执行。

流程引擎执行应用集群部署流程,开始进行应用集群部署,流程引擎顺序或并行地执行部署任务节点,部署任务节点可以调用应用集群模型相应节点的部署操作 API,通知模型节点执行部署操作。模型节点首先调用 OpenStack 的资源管理 API 创建虚拟机,为每个节点安装操作系统和 Docker 引擎,然后调用节点上 Docker 引擎的 API 完成集群节点软件镜像安装和加载。由于 Docker 镜像采用分层结构组织,上层的镜像可以共享下层的镜像,多个容器可以共用同一个只读的镜像,Docker 引擎能够自动识别 Docker 镜像的依赖关系,确保每个服务器都有所需

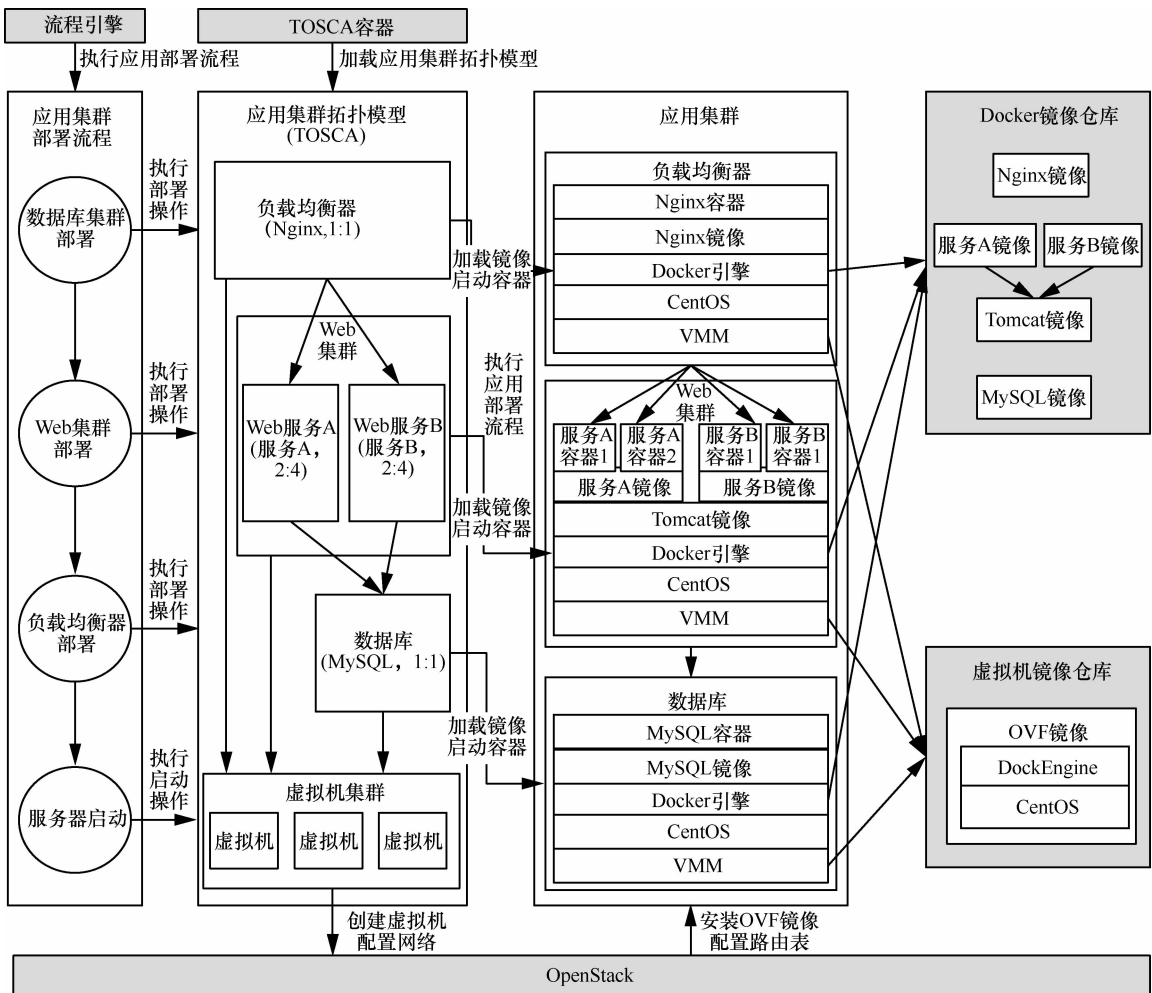


图3 部署平台核心部件协作关系

的软件镜像,且每个软件镜像层只下载和安装一次。

从上述分析可以看出,本部署平台的部署过程具有如下特点。

- 支持任意技术框架: Docker 镜像技术可以将任意的软件、类库、框架及其依赖关系打包成标准化的 Docker 镜像文件,统一了各种软件的构建、存储、分发、安装和运行过程。
- 支持任意拓扑架构: 可通过 TOSCA 模型灵活地编排和复用业务集群的拓扑结构。
- 支持全堆栈、全流程自动部署: 通过对接云资源平台和 Docker 平台,实现从硬件到软件的全堆栈安装配置,通过流程引擎和 TOSCA 容器驱动整个集群部署流程的自动化执行。
- 部署速度快、效率高: 轻量级的 Docker 镜像和容器技术降低了软件安装文件的存储和传输开销,秒级的 Docker 容器启动时间让业务平台快速就绪成为可能。

5 基于自动化部署平台的集群部署过程

利用互联网应用部署平台进行集群部署可分为 3 个阶段: 集群模板设计阶段,由有经验的架构师完成集群模板的设计; 集群模板实例化阶段,由团队的部署工程师将集群模板实例化为实际部署的模型; 集群模板部署执行阶段,由部署平台基于部署模型自动完成集群部署。各阶段的流程介绍如下。

5.1 应用集群模板设计阶段

有经验的架构师、系统工程师或管理员通过 TOSCA 模型设计器设计可复用的 TOSCA 应用集群模板,并将集群模板发布在自服务门户上。

5.2 应用集群模板实例化阶段

(1) 应用集群部署模型配置

开发者通过自服务门户选择一个合适的 TOSCA 应用集群模板,调整模板中的节点和关系,使之与要部署的应用集群架构一致,调整节点绑定的系统镜像和顶层 Docker 镜像,使之与要部署的集群节点一致。

(2) 集群部署包打包上传

通过自服务门户的集群打包功能将 TOSCA 模板文件和新增的本地镜像文件打包成 CSAR (cloud service archive) 部署包,通过管理门户上传 CSAR 部署包。

(3) 新增镜像保存与部署服务调用

门户解析 CSAR 部署包,将新增镜像保存在 Docker 镜

像仓库中,并将 TOSCA 模型文件发送给 PaaS 管理器,调用 PaaS 管理器的集群部署服务。

(4) TOSCA 部署模型解释

将 TOSCA 的集群拓扑模型解释成拓扑对象模型,保存在 TOSCA 容器中,将部署管理流程定义加载到 workflow 引擎中。

5.3 应用集群部署执行阶段

PaaS 平台 workflow 引擎执行集群部署流程,调用管理代理的接口依次执行虚拟机创建、网络配置、节点镜像安装、容器运行、容器配置等任务,自动完成整个应用集群的部署。

部署工程师可通过自服务门户手工执行完整的集群部署流程、子集群部署流程或者单个节点的部署操作。

6 结束语

基于 Docker、Cloudify 等开源 PaaS 技术构建的互联网业务自动部署平台,较好地平衡了部署效率与部署灵活性的问题,通过 Docker 节点支持任意的软件堆栈,通过 TOSCA/Cloudify 部署模型支持复杂的集群架构,实现了复杂互联网集群全堆栈、全流程的自动化部署,支撑互联网业务平台更好地应对用户需求的快速变化和用户规模的快速增长。

参考文献

- 1 Miglierina M. Application deployment and management in the cloud. Proceedings of the 16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania, 2014
- 2 孙宏亮. Docker 源码分析(一): Docker 架构. <http://www.infoq.com/cn/articles/docker-source-code-analysis-part1>, 2014
- 3 Sun H L. Docker source code analysis(I): Docker architecture. <http://www.infoq.com/cn/articles/docker-source-code-analysis-part1>, 2014
- 3 Cloudify 3.2 architecture overview. <http://getcloudify.org/guide/3.2/overview-architecture.html>, 2014
- 4 徐鹏, 陈思, 苏森. 互联网应用 PaaS 平台体系结构. 北京邮电大学学报, 2012, 35(1): 120~124
- 5 Xu P, Chen S, Su S. PaaS platform architecture for internet application. Journal of Beijing University of Posts and Telecommunications, 2012, 35(1): 120~124
- 5 张海瑞. 云平台自动化部署模块的设计与实现 (硕士学位论文). 哈尔滨: 哈尔滨工业大学, 2013
- 6 Zhang H R. Design and implementation of the automatic deployment module for cloud platform (master dissertation).



Harbin: Harbin Institute of Technology, 2013

[作者简介]



何震苇,男,中国电信股份有限公司广州研究院工程师,主要研究方向为容器技术、PaaS 平台、开源云平台等。



严丽云,女,中国电信股份有限公司广州研究院工程师,主要研究方向为虚拟化技术、开源虚拟机。



李慧云,女,中国电信股份有限公司广州研究院工程师,主要从事终端中间件技术、HTML5 技术研发工作。



张凌,女,中国电信股份有限公司广州研究院高级工程师,主要研究方向为网络功能虚拟化及业务编排。

陆钢,男,中国电信股份有限公司广州研究院高级工程师,主要研究方向为业务平台、能力开放、开源软件等。