

一种基于 Docker 的数据中心云平台实现方法及系统

A Cloud Platform for Data Center Based on Docker

中图分类号: TN929.5 文献标志码: A 文章编号: 1009-6868 (2017) 01-0060-03

摘要: 为解决现有平台即服务(PaaS)云平台的可用性和兼容性问题,提出了一种基于 Docker 的通用容器解决方案,并进一步提出基于该容器的云平台系统架构,详细描述了各个子系统及其功能模块的设计实现。认为通过引入动态调度模块,并复用多个分布式系统的公共组件,将使该系统架构的调度能力更加开放。

关键词: PaaS; Docker; 平台架构; 资源调度

Abstract: In this paper, a common container solution based on Docker is proposed to enhance availability and compatibility of platform as a service(PaaS) cloud platform. Furthermore, the cloud platform architecture based on Docker is proposed, and the design and implementation of its subsystems and functional modules are also described. It is believed that scheduling possibilities should be improved by dynamic scheduling module and common components of distributed system reuse.

Keywords: PaaS; Docker; platform architecture; resource scheduling

彭勇/PENG Yong¹
谢剑/XIE Jian¹
童遥/TONG Yao²
申光/SHEN Guang²

(1. 广西科技大学, 广西 柳州 545006;

2. 中兴通讯股份有限公司, 江苏 南京

210012)

(1. Guangxi University of Science and Technology, Liuzhou 545006, China;

2. ZTE Corporation, Nanjing 210012, China)

1 应用服务部署现状

随着信息化技术的深入发展,数据中心的应用服务数量急剧增加,原有采用实体机和虚拟机部署应用服务的方式面临着以下的问题:

- (1) 数据中心资源利用率不高。
- (2) 应用环境管理复杂,维护效率不高。
- (3) 应用服务迁移困难。
- (4) 传统应用服务部署启动慢。

2 现有 PaaS 云平台分析和 Docker 容器技术

2.1 现有 PaaS 云平台分析

平台即服务(PaaS)是云服务的

收稿日期: 2016-10-10
网络出版日期: 2017-03-02

一种^[1],服务提供商不仅提供按需索取的硬件和操作系统服务,还提供了应用程序平台和解决方案。通过 PaaS 平台,应用开发者可以很方便地把应用和服务托管在平台之上,而不用关心它们底层的硬件配置和运行环境。目前,业界已涌现出众多的 PaaS 云平台^[2],但它们均存在很大的不足。首先就是应用的运行环境兼容性较差,单一化的运行托管环境会导致应用的兼容性降低,不利于 PaaS 平台的快速扩张和通用性要求;其次, PaaS 平台内置的基础能力也不足,无法满足应用提供者对于能力的多样化需求^[3];另外,现有的 PaaS 云平台一般仅提供针对特定编程语言的应用运行环境,如 Google 应用程序工程(GAE)^[4]仅支持 Python 和 Java 运行环境,新浪应用程序工程(SAE)^[5]

仅支持 Java 和 PHP5 运行环境,因此急需提升平台的可用性和扩展性。

2.2 Docker 容器技术

Docker 是基于 Linux 的 LXC(Linux 容器)技术,把应用包装在容器里,使其具有移植性和独立性。具体来讲, Docker 提供标准化的 Image 和 Image 共享等功能,能够很简便地在 Docker 内部嵌入一个 Web 服务器(如 Jetty、Tomcat 等),更重要的是,使用 Docker 可以运行非常多的容器进程,针对不同 Web 服务能提供不同的 Image,并将各类不同的应用运行环境统一封装,对外提供统一管理和操作接口,提高 PaaS 平台的可用性和灵活性。

3 基于 Docker 的 PaaS 云平台设计

基于 Docker 的数据中心云平台的系统架构如图 1 所示,下面对该 PaaS 云平台涉及的主要实体和实体间的逻辑子模块进行描述。

(1) 云平台调度子系统。

云平台调度子系统是整个云平台的大脑,用于合理地分配系统资源,具体说来就是一个容器到底落地在哪些物理服务器上^[6]。调度子系统可细分为两个部分:初次调度和动态调度。初次调度是当用户第一次创建应用时进行的调度,包括资源的创建、删除、修改等操作;而动态调度是指在系统运行过程中的调度,主要是针对某些负载过高、过低的容器进行 scaling out、scaling in 的调度,实现资源的合理分配、整合。

(2) 路由控制子系统。

路由控制子系统在 PaaS 云平台中对所有进来的请求进行路由,该系统分为两层:第1层是从外部域名到内部临时域名的路由,第2层是从内部临时域名到容器的路由^[7]。提供内部临时域名是为了方便应用开发者在提交应用之后再次检查确认,路由控制子系统还提供负载均衡、健康检查、会话粘滞等高级功能。

(3) 通用容器子系统。

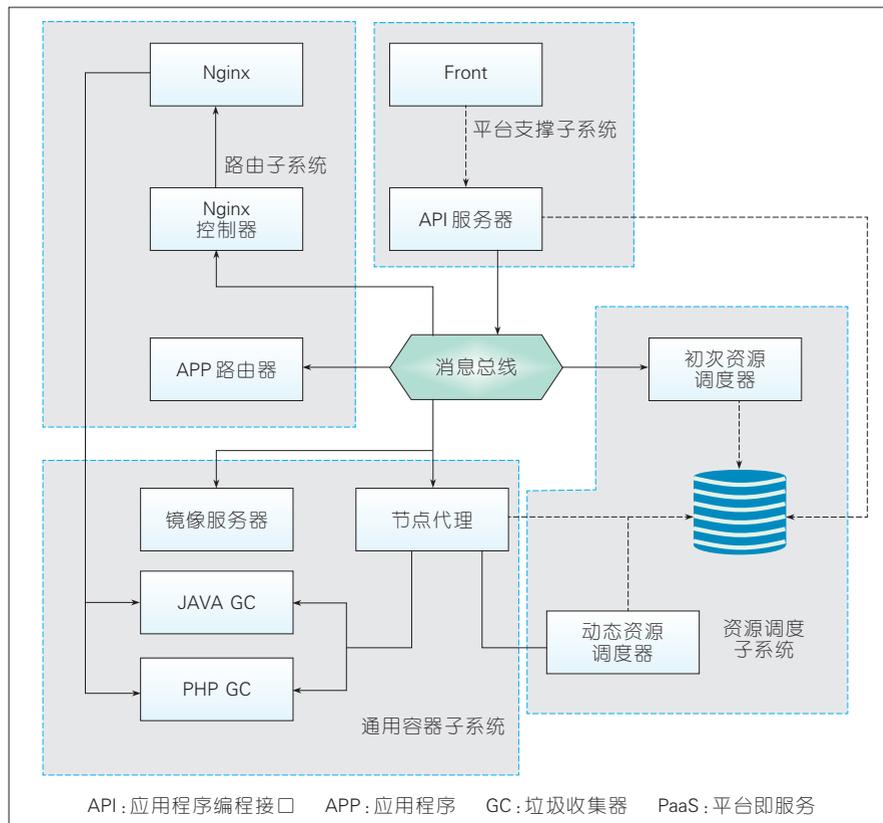
通用容器可抽象出两层,分别为适配层和容器层。适配层一方面可以作为驻留结点采集主机负载等信息,另一方面可以适配底层的容器。容器层则是在 Docker 应用程序编程接口(API)的基础上封装出不同的 Web 容器,比如 Docker-Jetty、Docker-Tomcat 等。同时还引入了镜像服务器这个集中式镜像管理模块。

(4) 平台支撑子系统。

平台支撑子系统是平台中的基础模块,提供基础能力的支持,比如:API 服务器将系统的能力统一抽象出 restful 的 HTTP 接口,Memcached 提供不同容器之间的 session 共享,Cache 提供高性能的分布式对象缓存服务,MySQL 提供原生的关系型数据库,Freedisk 提供小文件存储的服务。

4 基于 Docker 的 PaaS 云平台实现

本系统分为资源调度子系统、路



▲图1 基于 Docker 的 PaaS 云平台架构

由控制子系统、通用容器、平台支撑子系统4个子系统。

4.1 资源调度子系统的实现

资源调度子系统是整个系统的核心,当 PaaS 应用请求通过 API 服务器到达系统内部时,是由资源调度子系统决定数据流的走向。

对于调度子系统,我们在实现时又细分为初次调度和动态调度^[8]。

初次调度是指请求第1次到来时的调度,调度子系统在收到申请部署应用之后,通过该次调度决定承载该应用的多个容器分别落在哪几台物理服务器上。

而动态调度是指在系统运行过程中,通过对应用状态和容器状态的监测控制,对应用动态地进行扩容、缩容等。

资源调度子系统主要包括的模块如下所述。

(1)初次资源调度器:实现所有

初次资源操作请求的中转和协调,实现方式分为两步。第1步先取出所有的物理服务器,针对不同的请求策略使用不同的过滤器来过滤物理服务器,得到可用的服务器列表,保证主机的可用性。第2步是引入权重和打分概念,初次资源调度器通过不同的权重计算每个主机的得分,然后按照主机的得分从高到低排序,并按照应用部署多个副本(默认是2个)到得分高的主机上。

(2)动态资源调度器:实现动态调度,动态地增加、减少应用后端容器的数量。动态资源调度器获取到主机和 Docker 容器的负载信息后,按如下策略完成应用副本扩充:根据相应的调度策略选择负载最低的主机,在该主机上部署此应用的副本,同时向路由子系统发送请求,请求修改路由的指向,使得新加入的应用副本可接收请求,从而实现负载均衡。当集群中所有主机都接近于高负载状态

时,将停止应用副本的扩充,并向管理员发出告警邮件。同理,若发现某应用的所有副本的平均访问量过低,则按一定策略进行应用副本的收缩,此外,在副本扩充和收缩的过程中,添加定期碎片整理,在保证同一应用的多个副本分布在不同主机的前提下,尽量将分散的应用汇集到部分主机上。

(3)节点代理:一方面作为驻留进程运行在每个计算节点上,周期性地向初次资源调度器上报心跳数据,心跳数据包含CPU利用率、内存使用率和硬盘剩余空间等信息;另一方面适配底层的 Docker 容器,并获取其上每个容器的负载数据,包括CPU占用率、内存使用率、硬盘读写状态等。

4.2 路由控制子系统的实现

路由控制子系统维护了整个系统的路由信息,包括外部域名到内部临时域名和内部临时域名到容器(ip+端口)的映射,在底层通过对 Nginx 反向代理服务器的控制来实现 Web 请求的分发和比例控制。

路由控制子系统包括如下所述的主要模块。

(1)APP 路由器:整个 PaaS 云平台路由的核心控制组件,该组件负责维护 1 份包含所有应用路由策略的路由表,并根据特定的任务调度算法控制 Nginx 以完成对应用访问请求的路由,实现多个应用副本间的负载均衡、健康检查、会话粘滞等功能。该模块在接收到域名增加或删除操作的时候,先进行合法性判断,确定合法之后再判断是否需要更新路由表,如果需要则通知 Nginx 控制器进行路由表的更新操作。

(2)Nginx 控制器:跟 Nginx 同机部署。在收到 APP 路由器的更新路由信息的通知后,从数据库中加载最新的路由信息,并通过 Java 的 Velocity 技术生成对应的配置文件,替换当前 Nginx 的配置文件并重启 Nginx,从而使得路由信息及时生效。

(3)Tengine:在 Nginx 的基础上,针对大访问量网站的需求,添加了高级功能和特性,比如一致性 hash,会话保持,健康检查,根据服务器状态自动上线、下线等。

4.3 通用容器的实现

基于对 Docker 的封装,提供了多种 Web 容器,比如 Java 的容器、PHP 的容器等。另外,为了方便平台管理者快速发布、删除容器,还创新性地引入镜像服务器这个集中式镜像管理模块。

4.4 平台支撑子系统的实现

该平台还包括其他一些支撑组件,具体如下所述。

(1)API 服务器:整个系统对外提供的接口,使用 Spring 框架实现,把系统的能力通过 Restful 的 HTTP 接口暴露出来,方便系统管理员调用。

(2)Memcached:整个 PaaS 平台容器的内存共享池,通过在容器内部把 session 信息重定向到该共享内存池中,实现会话保持^[9]。

(3)Nginx-Log:分析 Nginx 的日志信息,采集每个应用访问量等信息。

(4)基础能力:比如 Mysql 数据库、定时任务等一些公用的服务。

5 结束语

针对现有应用服务部署存在的问题,以及现有 PaaS 云平台兼容性和扩展性的问题,我们提出了一种基于 Docker 的数据中心 PaaS 云平台实现方法。

未来可以使用容器逐步替换数据中心云平台中所有基础模块,如监测控制模块、负载均衡模块、服务管理模块、日志收集模块、消息中心管理模块等,使数据中心云平台计算、存储进一步分离。

参考文献

- [1] 吴朱华. 云计算核心技术剖析[M]. 北京:人民邮电出版社, 2011
- [2] 卜文军, 邹朝斌, 于佳琳. 热点云计算平台推荐[J]. 硅谷, 2011, 10(5):10-10. DOI: 10.3969/j.

issn.1671-7597.2011.05.010

- [3] 徐鹏, 陈思, 苏森. 互联网应用 PaaS 平台体系结构[J]. 北京邮电大学学报, 2012, 35(1):120-124. DOI: 10.3969/j.issn.1007-5321.2012.01.028
- [4] 李露. GAE 的技术发展及应用[J]. 科技创业月刊, 2012, 26(8):192-193. DOI: 10.3969/j.issn.1672-2272.2012.08.088
- [5] 丛磊. 新浪云计算 SAE 的技术演变[J]. 程序员, 2014, 15(9):90-95
- [6] NIVODHINI M K, KOUSALYA K, MALLIGA S. Algorithms to Improve Scheduling Techniques in IaaS Cloud[C]// ICICES 2013, USA:IEEE, 2013:246-250. DOI: 10.1109/ICICES.2013.6508188
- [7] 高允翔, 汤泳, 肖伟. 运营商私有云的 DaaS 构建方案初探[J]. 邮电设计技术, 2012, 55(10):20-22
- [8] VAQUERO L M, RODERO-MERINO L, BUYA R. Dynamically Scaling Application in the Cloud[J]. ACM SIGCOMM Computer Communication Review, 2011, 41(1):45-52. DOI: 10.1145/1925861.1925869
- [9] PETROVIC J. Using Memcached for Data Distribution in Industrial Environment[C]// ICONS 08, USA:IEEE, 2008:368-372. DOI: 10.1109/ICONS.2008.51

作者简介



彭勇,广西科技大学网络与现代教育技术中心工程师;研究方向为云计算、信息安全等。



谢剑,广西科技大学网络与现代教育技术中心工程师;研究方向为云计算、数据库、数据挖掘等。



童遥,中兴通讯股份有限公司工程师;研究方向为云计算、统一通讯等。



申光,中兴通讯股份有限公司工程师;研究方向为云计算、统一通讯等。