

# 基于 Docker 的平台即服务架构研究

Research of Platform as a Service Architecture Based on the Docker

张 建\* 谢天钧  
ZHANG Jian XIE Tian-jun

**摘 要** 平台即服务 (PaaS) 作为云计算的三种服务模型 (SaaS, PaaS, IaaS) 之一, 正受到学术界和企业界越来越多的关注。Docker 作为新兴的轻量级容器引擎为 PaaS 的构建提供了可靠的支持。本文探究了基于 Docker 的 PaaS 平台的架构模型, 对 Docker 关键技术进行阐述, 提出一种以 Docker 为基础的平台即服务架构模型 (DpaaS)。从而为企业搭建稳定、可扩展、高可用的 PaaS 平台提供参考。

**关键词** PaaS 云计算 Docker

**Abstract** Platform as a Service is one of the three service model (SaaS,PaaS,IaaS) about cloud computing. It attracts more and more attentions from academies and enterprises.Docker is a lightweight container engine, which provide reliable support.In the basis of studying Docker, the dissertation proposes a PaaS Model based on Docker (DPaaS) to help company build stable, scalable, and high available private cloud PaaS service.

**Key words** Docker PaaS Cloud Computing

doi : 10. 3969/j. issn. 1672 - 9528. 2014. 10. 60

云计算作为近些年学术界和企业界研究的热点领域, 正被越来越多的人所熟知和重视。按照云计算的服务模型划分, 云计算可分为软件即服务 (SaaS)、平台即服务 (PaaS)、基础设施即服务 (IaaS)。基于 PaaS 提供的平台运行时, 用户 (开发者) 可以便捷地开发和部署应用程序, 将应用程序托管在 PaaS 管理的云基础设施中, 从而节省大量的平台搭建和维护工作, 并达到缩短开发周期, 降低运维成本的目的<sup>[1]</sup>。Docker 作为一个容器引擎向 PaaS 提供了基础的资源隔离和标准化打包部署能力, 从而使得基于其上的 PaaS 平台构建简单高效。

## 1 研究背景

PaaS 主要受众就是 web 服务的开发者。开发者开发的 web 应用对外提供服务除本身的程序代码外, 还需要基础的语言运行环境, 存储空间等<sup>[2]</sup>。PaaS 为开发者提供了快速构建 web 服务的能力<sup>[3]</sup>。PaaS 主要向用户提供三种服务能力: 1. 基础的语言运行环境。比如谷歌的 GAE 支持 Python 和 Java 的应用部署。2. 服务能力 (存储、计算等) 的动态扩展。3. 通用基础服务的能力, 比如消息中间件, 分布式缓存等基础服务。通过这些服务

的提供使得开发人员将关注点集中在自身的业务需求上, 这对于开发人员快速提高开发效率具有重要意义<sup>[4]</sup>。Docker 的出现让开发和运维都变得简单, Docker 像集装箱一样将应用及其相关依赖进行整体打包, 移植到任何支持 Docker 的环境中运行。

## 2 关于 Docker

Docker 是一个基于 Linux Container 的 Container 容器引擎, 并遵从 Apache2.0 协议开源。目前 Docker 已经获得了包括 IBM、Google、RedHat 等公司的关注和技术支持。Docker 利用轻量级虚拟化技术进行资源隔离, 并可以将各种环境依赖和 web 应用一起打包, 并可方便的移植和重新部署。

### 2.1 Docker 和传统的虚拟化技术

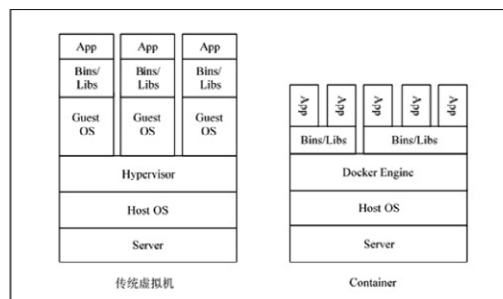


图1 Docker 和传统虚拟化技术对比

\* 北京工业大学软件学院 北京 100124

Docker 区别于传统的虚拟化技术。如图 1 所示, 传统的虚拟化技术如 vSphere 和 Hyper-V 目标是建立一个可以执行应用的整套操作系统的沙盒环境, 即虚拟机。而 Docker 使用的 Container 技术则是将应用所需要的相关源码、依赖库、环境配置等都打包起来建立一个沙盒执行环境。Container 采用公用操作系统的方式, 不需要安装 Guest OS, 所以 Container 拥有快速的启动速度, 方便建立和销毁。

## 2.2 Docker 解决的主要问题

### 2.2.1 复杂的环境依赖

Web 应用正常提供服务需要有很多环境依赖, 包括操作系统、中间件以及其他后端服务。这给应用的快速开发带来很多麻烦, 并使得开发团队增加很多不必要的困扰。Docker 通过将多种依赖打包成 image 的形式, 方便地进行环境移植和搭建, 并且 Docker 支持 image 的叠加。从而以一种搭积木的形式构建环境。

### 2.2.2 虚拟化成本

传统的虚拟化技术包括 KVM、Xen 等都是通过在 Host OS 上构建 Guest OS 来达到资源环境的隔离。但 Guest OS 本身就需要占用很大的资源。这对于资源是一种极大的浪费。Web 应用需要的仅仅是一个可用的完整运行环境, 而非整个操作系统。Docker 利用 LXC(Linux Container) 对系统资源进行隔离, 从而实现了类似虚拟机的隔离效果, 但是性能更加优秀。

### 2.2.3 应用的快速部署

Docker 将应用程序标准化, 将其进行整体打包后存储在 Docker Registry 中, 部署时只需要从 Docker Registry 中获取需要的 image 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, 从而大大简化了应用的部署成本。

## 2.3 Docker 的核心技术

### 2.3.1 隔离性

对于不同业务实例之间相互隔离, 一般的方式是采用虚拟机的手段进行业务间的隔离。但 Docker 基于 LXC, 采用 container 的方式进行隔离。主要通过内核的 namespace 将进程、网络、文件系统等隔离开。

### 2.3.2 资源可度量

cgroups(control groups) 是 Linux 内核提供了一种可以限制、记录、隔离进程组所使用物理资源的机制。其提供了类似文件的接口, 通过将数据写入文件的形式实现资源控制度量。cgroups 可以实现对 blkio、CPU、

cpuacct、cpuset、devices、freezer、memory、net\_cls、ns 九大子系统的限制。

### 2.3.3 便携性

AUFS 是 Docker 支持的一种文件系统。其可以将不同目录挂载到同一个虚拟文件系统下, 并可以为每一个成员目录设定 readonly、readwrite 和 writeable 权限。Docker 中具有 readonly 的文件称为镜像 (image)。AUFS 有一个分层的概念, 通常将 readonly 和 writeable 的层联合在一起, 从而在 readonly 层不变的基础上对 writeable 层进行写操作。Docker 利用 AUFS 的这种层级关系可以将多个具有依赖关系的 image 组合在一起。有了层级化的 Image 做基础, 理想中, 不同的 APP 就可以既可能的共用底层文件系统, 相关依赖工具等, 同一个 APP 的不同实例也可以实现共用绝大多数数据, 进而以 copy on write 的形式维护自己的那一份修改过的数据等。

## 2.4 Docker 主要核心组件

Docker 整体采用 C/S 架构, 其主要核心模块包括 Docker Client、Docker Daemon、Docker Registry 以及 Docker Container<sup>[5]</sup>。

### 2.4.1 Docker Client

Docker Client 是用户用来和 Docker Daemon 建立通信的客户端。Docker Client 发出的请求由 Docker Daemon 进行处理。通过使用 Docker 命令行工具发起请求, 从而对 Docker Container 进行管理。Docker Client 支持多种通信方式包括 tcp、unix socket 等, 此外还支持安全传输层协议 (TLS)。

### 2.4.2 Docker Daemon

Docker Daemon 是 Docker 中一个常驻后台的模块, 其主要包括两个部分, Docker Server 和 Docker Engine。Docker Server 主要负责接收 Docker Client 的请求, 将请求分发给相应的 handler 进行处理。Docker Engine 是整个 Docker 的核心执行组件。其将每一次具体的操作抽象为一个 job 的执行, 比如说一个容器的创建过程。job 类似于操作系统中的进程的概念。

### 2.4.3 Docker Registry

Docker 是一个存储容器镜像的仓库。通过镜像的加载可以创建容器的初始化文件目录。Docker Registry 支持镜像的搜索、下载和上传功能。Docker Registry 可以使用官方的公有仓库 Docker Hub, 也可以搭建自由的私有 Docker Registry。

## 2.5 Docker 工作流

Docker 的强大之处在于其轻量级的资源需求和灵活的可移植性。一个 Docker 的典型工作流如图 2 所示：

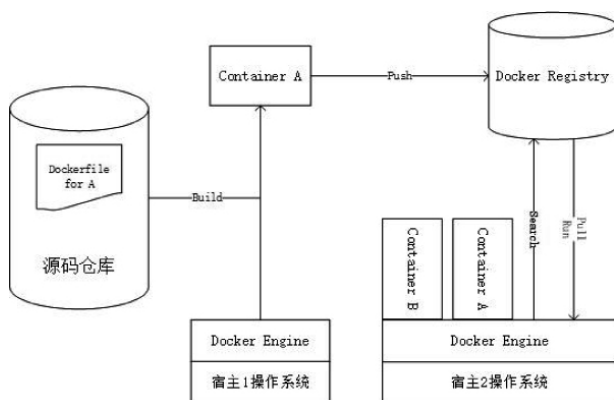


图 2 典型 Docker 工作流

Dockerfile 描述了一个镜像的构建过程，包括环境变量的设置、依赖软件的安装和启动、暴露的服务端口等。通过 Dockerfile 构建镜像后通过 push 指令将镜像上传到 Docker Registry。部署有 Docker Engine 的其他宿主只需要搜索 Docker Registry 中自己需要的镜像，就可以将其整个下载 (pull) 下来。完整的镜像包含了应用运行的所有依赖，所以在新的宿主系统上应用可以直接对外提供服务，不需要再进行安装依赖软件、配置环境变量等工作。

## 3 基于 Docker 的 PaaS 架构

基于 Docker 的 PaaS 平台 (DPaaS) 包括控制层和运行层两大核心部分以及相关的接入层、通用服务、平台监控、日志处理几大辅助模块，平台整体的架构图如图 3 所示。

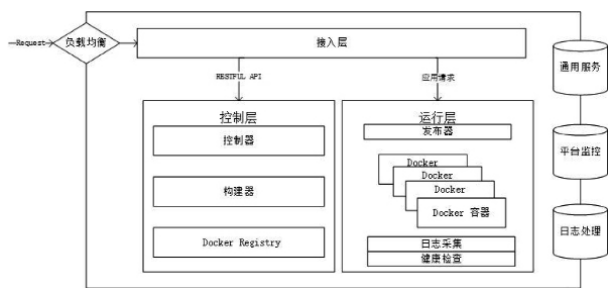


图 3 基于 Docker 的 PaaS 架构

### 3.1 接入层

所有的请求通过负载均衡模块后转发给接入层，接入层将请求区分为平台的自身的 RESTful API 调用以及普通用户的服务请求。RESTful API 会被转发给平台控制

层进行处理，而用户的应用请求则直接路由给了运行层，由运行层中的 Docker 容器提供服务。

### 3.2 控制层

控制层是整个 PaaS 平台的控制核心，其负责整个平台的控制调度、流程管理、镜像构建与存储等功能。控制器负责处理控制请求，发消息给其他模块，协调整个平台组件间的协同运行。构建器主要负责镜像构建的具体工作，并和 Docker Registry 进行交互，实现镜像的存储调度。

### 3.3 运行层

运行层实际部署 Docker 容器，对外提供各种 Web 服务。接入层将不同服务的应用请求分发给暴露服务端口的 Docker 容器，Docker 容器实际处理各种应用请求。Docker 容器通过向发布者注册自己的服务信息，从而使得接入层可以发现服务，并向其发送应用请求。日志采集组件采集 Docker 容器运行过程中产生的各种日志，并将其传输给日志处理模块。健康检查模块检测 Docker 容器的运行状态，将其健康信息定期发送给监控模块。

### 3.4 通用服务

通用服务是平台向平台使用者提供的各种公共服务组件，比如数据缓存服务、消息队列服务、URL 抓取服务等。这些公共服务有平台托管，用户只需按需计费即可。通用服务使得应用的开发者只需要将经历完全关注在自身业务逻辑的实现上，而无需关系其他。

### 3.5 平台监控

平台监控负责整个运行层的状态监控。监控信息由健康检查组件提供。通过平台监控，平台可以实现应用的动态扩展，根据容器负载情况动态的创建删除服务容器，下线问题容器等。

### 3.6 日志处理

日志处理模块将日志采集模块传输到的日志进行分类、重建、存储以供其被其他数据挖掘服务使用。

## 4 结论

随着云计算技术的发展，IaaS 和 SaaS 都有了长足的发展，而 PaaS 由于公有 PaaS 限制过多、使用复杂、迁移不够灵活的特点，使其并没有 IaaS 和 SaaS 普及。本文在分析 Docker 相关技术的基础上，提出了基于 Docker 的轻量级私有云 PaaS 架构，为 PaaS 平台的设计提供了参考。

# 医院信息化与医院档案管理现代化现状分析

张轶森<sup>\*</sup> 杜鹏飞<sup>\*\*</sup>  
ZHANG Yi-sen DU Peng-fei

**摘要** 医院档案是医院发展全过程的见证者,其权威性是无可替代的。尤其是珍贵的历史资料,一旦丢失或者损毁,其后果是不可估量的。随着我国经济社会的迅猛发展,网络技术也得到了相应普及,社会资源网络化已经是不可阻挡的发展趋势,医院档案管理现代化不仅在一定程度上提升了工作效率,而且为科学的制定管理计划创造了前提条件。本文针对当前我国医院档案管理现状进行分析,试图在探讨过程中找出两者间存在的联系,以更好的适应医院档案管理信息化需要,从而促进医院的整体发展。

**关键词** 医院信息化;档案管理;现代化

doi: 10.3969/j.issn.1672-9528.2014.10.61

实现档案信息化管理是网络时代档案工作的最终目的,在传统的档案管理过程中都是完全依靠于管理员的责任心、管理水平及能力等方面,或多或少会与实际存在一些差异,以信息化进行医院档案管理,为传统档案管理注入了活力,信息化管理是通过编制相关程序对医院信息进行系统化整合、管理,方便查询,从而使得档案管理工作效率得到提高,这是传统的档案管理所无法比拟的<sup>[1]</sup>。随着医院改革的逐渐深入和竞争日益激烈,如何对医院档案进行有效管理是当前所面临的一大难题。要想立足于社会竞争就必须强化档案管理,提高档案资料的准确性,为提升档案服务质量提供前提条件。

\* 承德市中医院 河北 承德 067000

\*\* 承德市中心血站 河北 承德 067000

## 1 医院信息化发展的现状分析

在中国医院协会的带领下,医院信息化管理系统已经运用于我国大部分三级医院的档案管理中,其他基层医院的管理系统也在不断地发展、建设中。可以说,医院信息化在三十年的发展中不断成长,取得了重大突破和显著成就,图表1所示就是我国医院信息化投入比例。



图表1 我国2007~2015年医院信息化市场规模及增长率(2015年为预测值)

## 参考文献:

- [1] 林琳,滕腾,李伟彬.PaaS的范畴及架构标准化研究[J].信息技术与标准化,2012(10):25-32.
- [2] 姜文周,马明丽,李先毅.基于Cloud Foundry的PaaS云平台的设计与实现[J].微型机与应用,2014,33(2):60-62.
- [3] Walraven,Stefan,Truyen,et al.Comparing PaaS offerings in light of SaaS development:A comparison of PaaS platforms based on a practical case study[J].Computing,2014,96(8):669-724.
- [4] Garcia-Garcia.Andres,De Alfonso,et al.Overview of current commercial paas

platforms[C]//ICS0FT2011- Proceedings of the 6th International Conference on Software and Database Technologies.Seville,Spain:INSTICC Press,2011:368-376.

- [5] 孙宏亮.Docker源码分析(一): Docker架构[EB/OL].[2014-09-25].<http://www.infoq.com/cn/articles/docker-source-code-analysis-part1>

[作者简介] 张建(1966-),男,博士,副教授。研究方向:信息服务与云计算。

谢天钧(1988-),男,硕士在读。研究方向:信息服务与云计算。

(收稿日期:2014-11-15)