



专题:大数据

DC/OS 关键技术与应用场景

张基恒¹, 李大中², 张呈宇¹, 魏进武¹

(1. 中国联合网络通信有限公司研究院, 北京 100032;

2. 中国联合网络通信有限公司信息化事业部, 北京 100032)

摘要: 首先对数据中心操作系统内的主要技术, 包括 Mesos、Yarn、Kubernetes 和容器进行了阐述。根据国内外电信运营商的现状和面临的问题, 提出了一种解决方案, 并对技术实现及选型建议进行了讨论。最后对电信运营商应用 DC/OS 关键技术的案例进行了简要介绍。

关键词: 数据中心操作系统; 资源调度; 开源

中图分类号: TP316

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2016313

Key technologies and application scenario of DC/OS

ZHANG Jiheng¹, LI Dazhong², ZHANG Chengyu¹, WEI Jinwu¹

1. China Unicom Research Institute, Beijing 100032, China

2. China Unicom Group Information Division, Beijing 100032, China

Abstract: Firstly, critical technologies in data center operating system (DC/OS), including Mesos, Yarn, Kubernetes and Docker were presented. Based on the current situations and problems that both domestic and foreign telecom operators faced, a simple resolution was proposed and technical realization as well as the selection of techniques were discussed. At last, several cases of DC/OS application for telecom operator were introduced.

Key words: DC/OS, resource allocation, open source

1 引言

随着越来越多的平台、架构、系统、组件和工具等通过开源社区、小微型初创项目而衍生, 不再由 IT 或互联网巨头独自垄断, 这为 IT 领域各项技术的快速、活跃发展提供了良好的温床, 开源技术的迅速发展、活跃的社区力量, 使得一些理念如 DC/OS (数据中心操作系统) 以及技术如 Mesos、Kubernetes 和容器等应运而生, 并广泛地得到了实验和迭代改进。

2 DC/OS 关键技术

DC/OS 问世之前, 如谷歌 (Google) 等互联网企业就内部资源调度和分配、负载, 已逐步展开了探索, 形成了一些如 Borg 等的生产化项目。而在 2014 年, 由 Mesosphere 实现了第一个可进行统一调度所有数据中心及云资源的软件堆栈, 并称为 DC/OS。目前已投入实际生产使用的 DC/OS 包括 Google 的 Borg/Omega 系统和推特 (Twitter)、苹果 (Apple)、网飞 (Netflix) 等公司基于 Mesos 构建的系统。而



在2016年4月,Mesosphere将自己的DC/OS代码进行了开源,其中包含超过30项组件技术,最为典型的有Apache Mesos与Marathon。下文将对可用于DC/OS构建的开源技术包括Mesos、Yarn、Kubernetes和Docker进行简要阐述。

2.1 Mesos

Mesos是开源资源统一管理和调度平台,其实现逻辑如图1所示,由美国加州大学伯克利分校AMPLab开发,后应用于Twitter、Apple、Netflix等企业。其亮点在于两层调度的架构可以做到“大集群”、减小“中心节点”的压力以及在同一个集群中可接入多种应用、框架,提高分布式集群的利用率。

由于Mesos上的元数据可通过各个计算节点重新注册而进行重构,因此可通过Zookeeper解决单点故障问题。轻量级的Mesos架构包括以下4个组件。

(1) Mesos管理节点

系统的核心,负责管理接入Mesos的各个框架和计算节点,并将计算节点上的资源按某种策略(默认使用DRF(dominant resource fairness,主导资源公平)算法)分配给不同的框架。

(2) Mesos计算节点

接收并执行来自管理节点的命令、管理节点上的任务并为各个任务分配资源。

(3) 框架

计算框架,以注册的方式接入Mesos以便其进行统一

管理和资源分配。

(4) 执行器

主要用于启动内部任务。

2.2 Yarn

Yarn是新的Hadoop MapReduce框架,实现思路如图2所示。其主要为了解决Hadoop1.0扩展性较差、不支持多计算框架的问题,本质是一个资源管理架构。Yarn减小了JobTracker的资源消耗,并且让监测每一个工作子任务状态的程序更安全、更分布式化;对于资源的表示以内存为单位,较旧版本更合理。其亮点在于可以更快地进行MapReduce计算,实现对多架构的支持以及架构升级的简便性。

较1代MapReduce的设计思路,Yarn将JobTracker两个主要的功能,即资源管理和任务调度/监控,分解为独立组件资源管理器和应用管理节点。资源管理器全局管理所有应用程序计算资源的分配,每一个应用的应用管理节点负责相应的调度和协调。

2.3 Kubernetes

Kubernetes是Google多年大规模容器管理技术的开源版本,其实现思路如图3所示。其亮点在于具有容器编排能力、相对轻量级及易于管理运行Docker的容器。Kubernetes使用Docker对程序进行包装、实例化、运行,并以集群的方式运行、管理跨机器的容器,同时解决了Docker跨机器容器之间的通信问题。

Kubernetes框架包含以下组件。

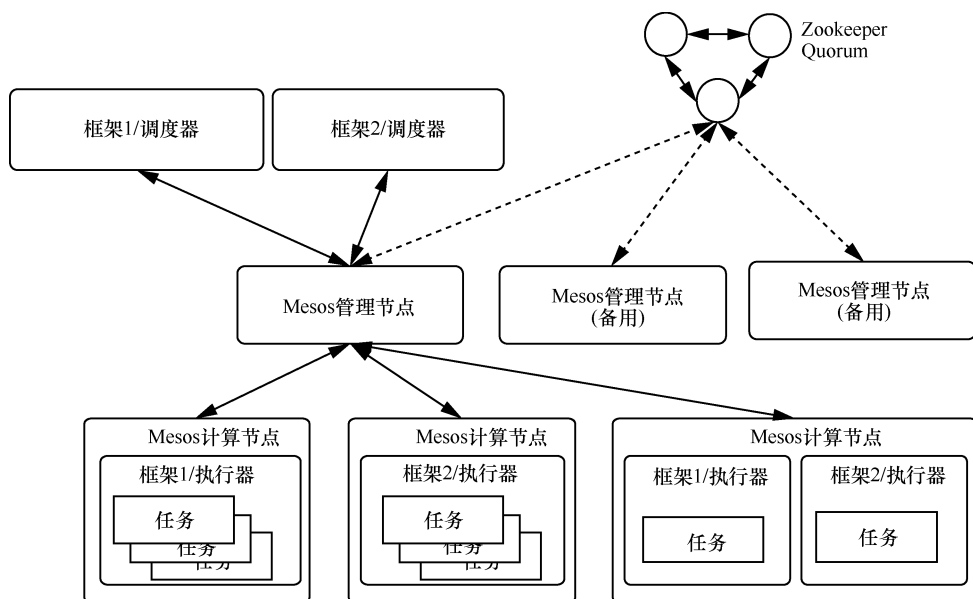


图1 Mesos实现思路

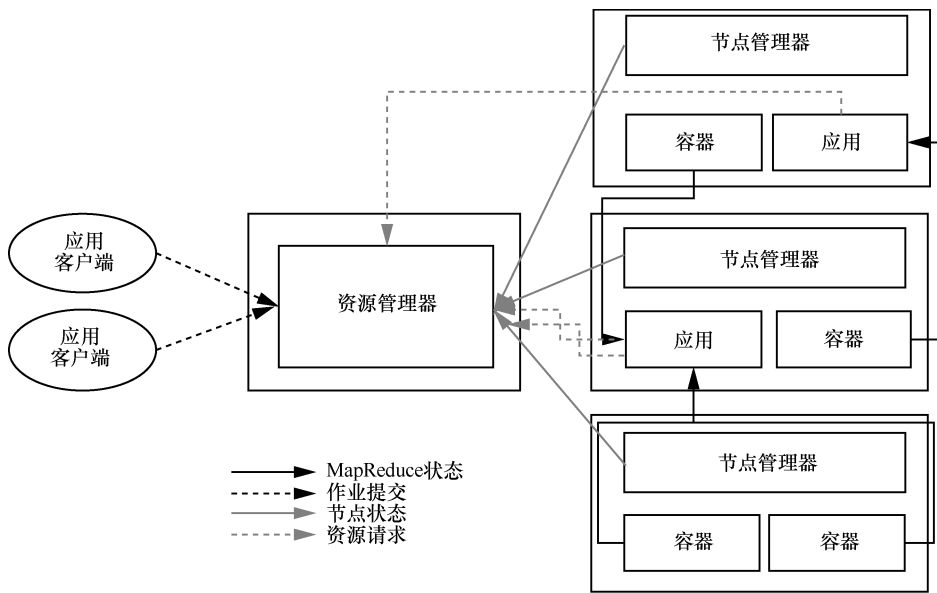


图 2 Yarn 实现思路

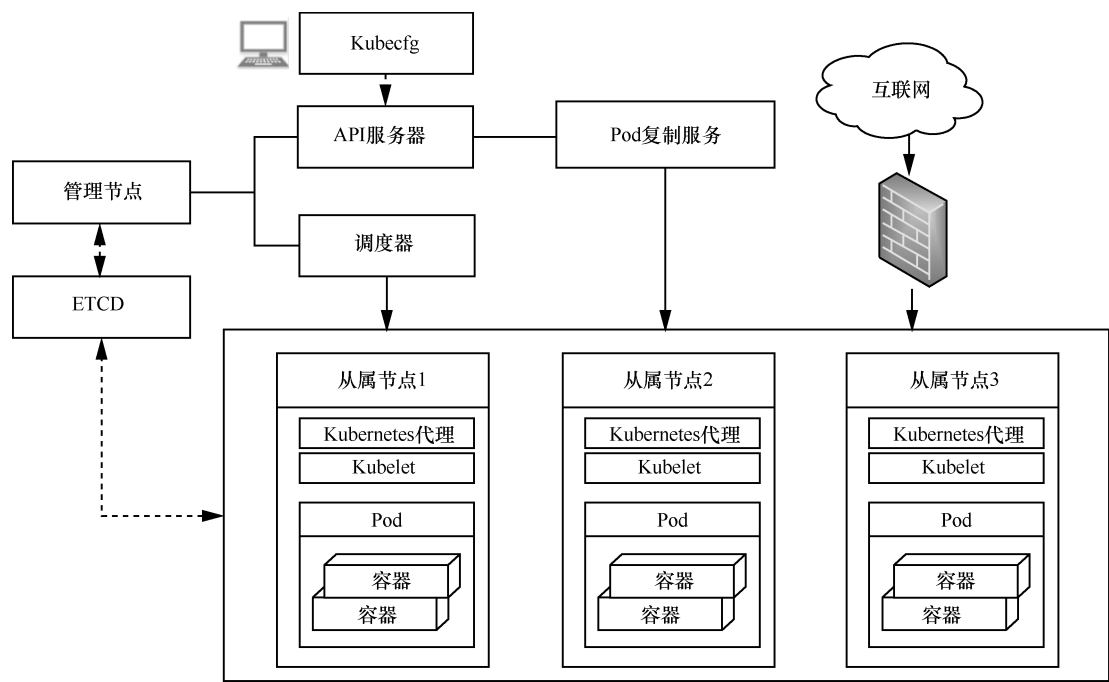


图 3 Kubernetes 实现思路

- Kubernetes 管理节点:管理整个系统。
 - Kubernetes API 服务器:系统入口,封装了核心对象的操作。
 - Kubernetes 调度器:负责集群资源调度,为新建的 Pod 分配机器。
 - Kuberlete 节点:运行节点,用于运行管理业务的容器。
 - Kuberlet:负责管控容器,从服务器接收命令。
 - Kubernetes 代理:负责为 Pod 创建代理服务。
 - Docker:Kubernetes 节点是容器运行节点,需要运行 Docker 服务。
- 首先,Kubecfg 将请求如创建 Pod 等,发送给Kubernetes 应用客户端,并由其将该请求发送给 API 服务器。API 服务器根据请求类型(如创建 Pod 时存储类型是 pods)并依此选择何种 REST 存储 API 对请求作出处理。REST 存储 API 对请求做相应的处理,并将处理结果存入高可用键值



存储系统(ETCD)中,在 API 服务器响应 Kubecfg 的请求后,调度器会根据 Kubernetes 应用客户端获取集群中运行的 Pod 及从属节点的信息。依据从 Kubernetes 应用客户端获取的信息,调度器将未分发的 Pod 分发到可用的从属节点上。

2.4 Docker

Docker 是一个开源的应用容器引擎,为开发者提供一个可将应用及依赖包打包至一个可移植的容器环境中,并发布到任意流行的 Linux 机器上。容器采用沙箱机制,容器间不存在接口。其亮点在于快速便捷的单次构建+导出运行模式及愈加丰富的生态系统。

Docker 采用 C/S 模式,使用远程 API 来管理和创建 Docker 容器。Docker 容器通过 Docker 镜像来创建。容器与镜像的关系类似于面向对象编程中的对象与类。Docker 采用 C/S 架构 Docker Daemon 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。Docker Daemon 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 Daemon 交互。

3 电信运营商 DC/OS 实现思路

3.1 现状与问题

云计算、大数据相关技术的普及推进了企业云化的建设和发展。虽然大部分国内外电信运营商已实现了 IaaS 资源池化和部分 PaaS 及应用资源池化,能够满足生产供给和部署。但大部分运营商的数据中心采用分隔式集群实现思路,包括以开源 Yam 或定制版大数据资源调度,承载数据类作业的集群以及数据库、缓存集群等;缺乏迫切资源的跨域调度能力;同时目前的运营系统还存在的资源运行

不均衡、时段运行不均衡等问题,具体包括:

- 应用难以实现灵活、快速地部署,应用系统的开发过程从开发到生产有不同的部署环境,对应的实现代码也面临着不同环境的开发和部署;
- 系统弹性伸缩能力不足,扩缩容需经过较多环节,无法实现快速、自动的伸缩,同时会造成资源的浪费;
- 现有资源利用率在 15%左右,无法充分实现资源的有效利用;
- 应用系统仍旧面临“烟囱式”的建设等现实问题,应用与平台紧耦合,高可用、有效监控、智能运维难以标准化。

鉴于此,电信运营商需设计、研发一种架构,以解决资源调度等现实问题并提供微服务化的实现环境。

3.2 功能结构

一种可满足电信运营商需求的 DC/OS 的功能实现如图 4 所示,分为 4 部分内容,分别是业务调度层、应用服务层、资源管理层以及自动化运营工具。

业务调度层的主要功能包括集群管理、服务接入、服务命名、服务发现、服务目录管理、负载均衡、服务访问控制等。

应用服务层主要负责容器集群的管理和大数据集群的管理。

资源管理层的主要功能包括底层计算资源的管理以及对其上注册的任务的管理。

自动化运营工具包括监控和多租户管理两大功能,其中监控又细分为对资源、服务、容器、集群的监控。

3.3 技术实现框架

为实现如图 4 所示的功能架构,需要引入资源调度技术、容器技术等,在技术路线选择上,则以社区领域的开源技术为基础形成自有版本,并兼顾开源社区本身的发展。

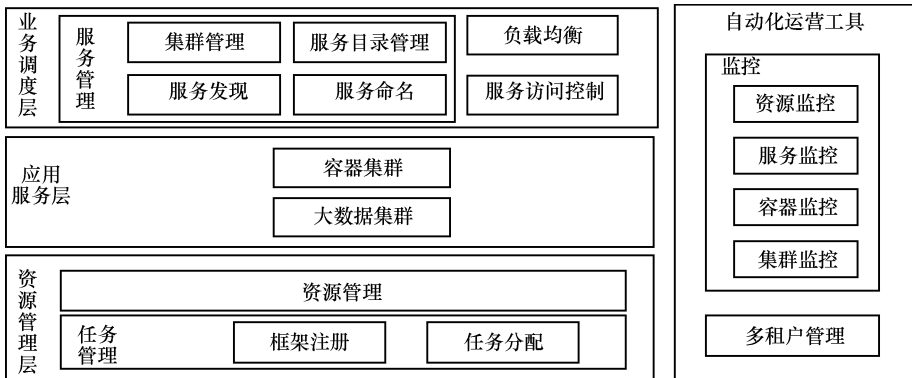


图 4 总体功能架构

一种较为适合运营商的 DC/OS 技术架构如图 5 所示。

3.3.1 Mesos 版本选型

开源 DC/OS 和商用 DC/OS 对比见表 1。

选型建议：可使用开源技术和自行研发结合方式实现，其中对资源的统一管理和调度采用开源的 Mesos 或者开源的 DC/OS 来支撑，从对比来看开源 DC/OS 与

ApahceMesos 核心模块功能一致,开源 DC/OS 增加了如框架的自动安装部署和门户等附加功能。

3.3.2 Kubernetes 和 Marathon 框架整合

Kubernetes 和 Marathon 对比见表 2。

选型建议：Kubernetes 和 Marathon 各有优点，且各自都有固定的生态圈,因此目前阶段无法选定对容器的编排

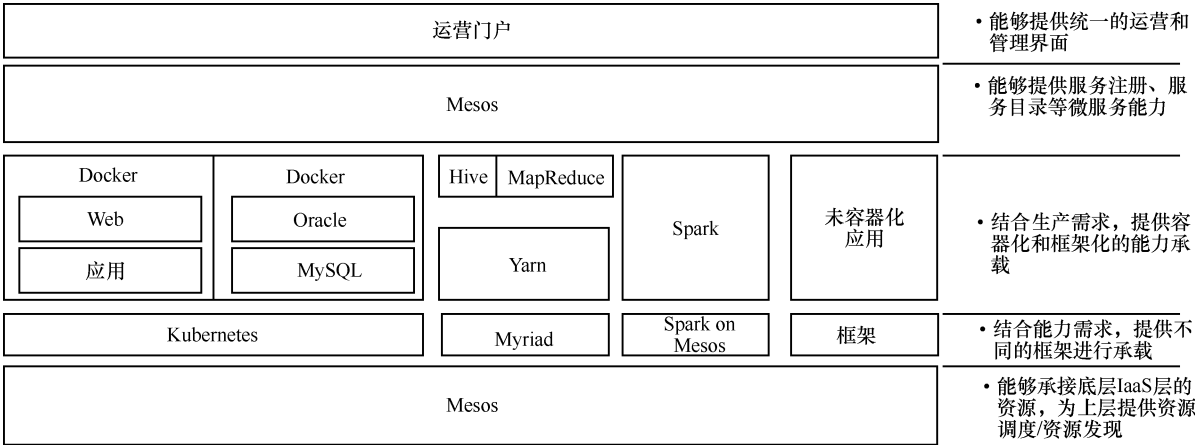


图 5 DC/OS 技术实现框架

表 1 开源 DC/OS 和商用 DC/OS 对比

类别	开源 DC/OS	商用 DC/OS
内核	Apache Mesos 0.28.1(Apache 原生版本无修改)	Apache Mesos 0.28.1(Apache 原生版本无修改)
功能	资源调度、任务监控、服务注册、服务路由、负载均衡、权限管理、图形界面、监控、框架私库、日志服务	服务注册、服务路由、负载均衡、权限管理、图形界面、监控、框架私库、日志服务、部署升级、网络管理、持续化集成、大数据套件、持久卷、容器自动扩容
生态	Mesos、Docker、Marathon、Frameworks (Myriad、Spark、Kafka···)	Mesos、Docker、Marathon、Frameworks(Myriad、Spark、Kafka···)
匹配度	强,DC/OS 除了提供开源 Mesos 的全部功能外,还提供一定的上层功能,包括服务治理、负载均衡、权限管理、框架私库等	弱,Mesos 所有节点运行在 Docker 上,然后用 Swarm 来管理这些 Docker 集群。这种架构下 Mesos 上的框架也必须运行在 Docker 里,对于 Myriad 存在存储、网络以及管理上的问题
前景	开源,随着 DC/OS 的发展,会有越来越多的厂商提供在 Mesos 上运行自身产品的框架,开源 DC/OS 本身无缝兼容开源 Mesos 上的框架,目前已超过 40 种	非开源,依赖于商业公司独立支撑,同时在开源 Mesos 上运行的框架不一定能直接运行在 Linker DC/OS 上

表 2 Kubernetes 和 Marathon 对比

类别	Kubernetes	Marathon
功能		
灰度发布	已经支持 (Rolling-upgrade)	部分支持(还需要手动操作)
容器编排	支持 Pod 方式	不支持 Pod 方式
容器安全	支持现有的容器安全机制比如 SELinux 等	仅支持 Linux 用户这种初步的安全机制
证书管理	提供对 Docker Hub 等证书的管理机制	不支持
持久化卷管理	支持(PV、PVC)	部分支持,相应的调度策略还在开发中
多机群(容灾)	支持	不支持
生态	Mesos、Docker	Mesos、Docker
匹配度	强,Kubernetes 在容器调度方面提供的功能更为完善	强,Marathon 近些年随着 DC/OS 的发展,功能也在向 Kubernetes 逐渐拉齐,另外 Marathon 可以通过进程调度的方式直接部署一些非 Docker 化的应用
前景	开源软件,Kubernetes 由谷歌大力推动,另外在社区和企业里惠普等公司也在推广,有广泛的商业应用	开源软件,Marathon 主要是由 Mesosphere 在进行推广,有广泛的商业应用



和调度管理采用拿一个开源工具,因此在 DC/OS 平台需要长期跟踪和集成两个容器编排工具。

3.3.3 对大数据服务框架的集成和封装

通过 Myriad 实现大数据服务框架的集成。

4 电信运营商 DC/OS 案例分析

与 DC/OS 相关的 Mesos、容器等技术在电信运营商领域获得了广泛的探索、研发与实践。

4.1 浙江移动 DC/OS 实现“双 11”1 折充值秒杀

中国移动以开源技术 Mesos、Marathon、Docker、HA 代理为引擎,在其上完成了自有数据中心操作系统 DC/OS 平台的设计、测试验证,平台采用 93 个主机节点,其中平台部分由 5 个节点构成 Mesos 管理节点集群,8 个节点构成 HA 代理集群,计算节点由 80 个 Mesos 计算节点组成。为验证其对业务的承载能力,中国移动选取浙江移动手机营业厅作为试点,将其手机营业厅系统迁移到 DC/OS 平台上,用于重点解决秒杀场景。并在 2015 年 11 月 11 日,推出手机营业厅充值 1 折秒杀的活动,借以实验和测试平台对其业务的支撑。在当晚 8 点,DC/OS 平台充分展现了其在应对业务高峰时的能力优势,顺利通过“双 11”考验。

数据显示,“双 11”活动期间,浙江移动手机营业厅系统承受的并发数峰值接近 6 万次/s,后台服务调用成功率均在 99.95%以上,成为浙江移动首个在单日实现 10 亿级 PV 的业务系统,其平台 CPU 利用率从资源池的 10%提高到 40%~50%。

随后,中国移动陆续将 CRM 核心业务逐步迁移,在 2016 年 4 月底 DC/OS 已成功承载其 16 套应用系统,涵盖

CRM 营业厅、手机营业厅等多套核心系统。

4.2 Verizon 的 DC/OS 支撑自有系统和应用

Verizon 公司作为国际电信运营商巨头之一,不仅重视电信运营商基础业务如语音及数据服务,同时也提供面向用户的应用托管业务以及 Verizon 内部服务如云服务、用户托管应用等。而这几年 OTT 的兴起,令传统通信业务逐渐减少,也令 Verizon 公司对其他非电信业务更加重视。但其传统的电信业数据中心架构仍是集群化的、计算资源孤岛式的,无法实现应用的自动化部署、扩缩容,因此需要一种较有限的集群管理工具,并可以实现隔离、灰度升级、打包、资源弹性分配等能力。

鉴于此,在 2013 年,Verizon 为改变计算和存储资源的低利用率和随之而来的运营低效率问题,开始选择 Docker 容器技术以及用来管理 Docker 容器及服务器集群的 Mesos 技术重构其基础设施架构,以支撑 Verizon 网络上数以万计的工作任务。2014 年底,Verizon 建立了一种以 Linux 为核心,由遍布数据中心的普通服务器构成的集群。在应用了新技术后,在硬件资源和应用部署两方面均有有效的提升。具体如图 6 所示。

(1) 硬件资源

- 资源利用率提升:其数据中心的资源利用率由最初的 10%~20%提高到 50%以上。
- 提升集群使用有效性:参考集群的历史数据,哪些应用上云或者从云上下线了,从而将硬件采购做得更合理。
- 减少硬件资源规划计划:闲置的集群资源可提供给新业务,因此无需为新业务规划预期规模所使用的

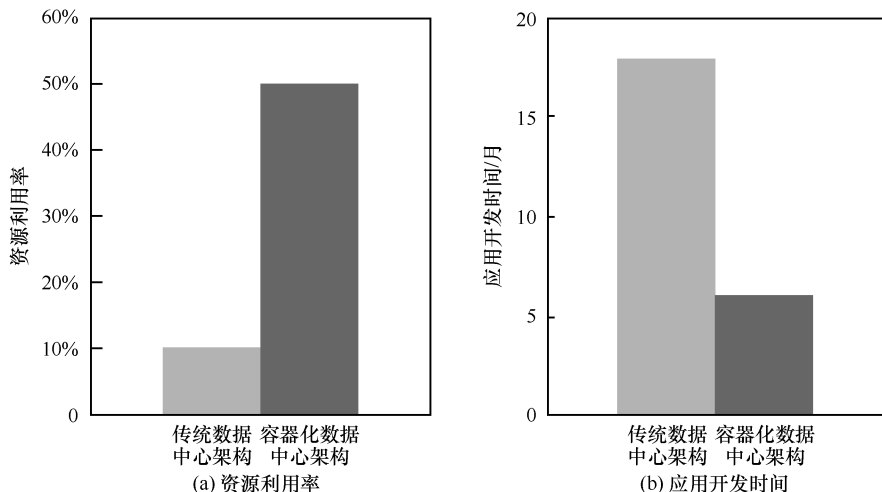


图 6 Verizon 传统/容器化数据中心效率

资源;进一步可根据所有应用的运营情况来增加硬件投入,按季度扩展集群规模。这种方法还可以使计划更具体和有效。

(2)时间成本

- 应用快速部署:支持在 72 s 内部署 50 000 个 Docker 容器,应用集群部署与过往相比提高了一个数量级。
- 开发成本降低:无需关注机柜和硬件环境,应用实现短平快的开发。

Verizon 计划在其 Mesos 集群上运行一些服务,第一批将被迁移到 Mesos 集群上的服务包括无线网络支撑系统和一些移动应用的后台以及 FiOS 网络支撑系统(光纤到户)。Verizon 还计划将其 Hadoop 和 Spark 分析任务从它们的专属集群上迁移到 Mesos 集群。其最终目标是希望该集群可达到如互联网巨头般的由廉价、单一的硬件设备组成的数据中心。

5 结束语

未来电信行业将会是一个 IT/CT 融合,电信运营商互联网化的趋势,IT 所占比重会越来越来大,电信行业的盈利模式也会从传统的提供通道到提供 IT 资源、提供的服务方向转变。近几年随着大数据、云计算、容器化、微服务、平台战略等新技术和新概念的层出不穷和快速发展,在业务支撑、架构能力、平台扩展性等方面对旧有的烟囱式建设的业务支撑系统提出了巨大的挑战。因此对现有业务和设备进行容器化改造,对数据中心乃至全网 IT 系统进行 DC/OS 改造,则成了一个必然的选择。

目前,DC/OS 相关技术已在部分运营商有成功使用的案例,证明相关技术在运营商的业务和网络中具有不可忽视的作用,同时运营商也需从自身定位和转型战略的视角来看待 DC/OS 的发展,并选择自身合适的业务分步进行改造和迁移。

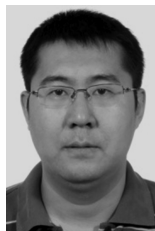
参考文献:

- [1] HINDMAN B, KONWINSKI A, ZAHARIA M, et al. Mesos: a platform for fine-grained resource sharing in the data center[C]// The 8th USENIX Conference on Networked Systems Design and Implementation, March 30–April 1, 2011, Boston, MA, USA. New York: ACM Press, 2011: 429-483.
- [2] VAVILAPALLI V K, MURTHY A C, DOUGLAS C, et al. Apache Hadoop YARN: yet another resource negotiator [C]// Symposium on Cloud Computing, May 27–June 1, 2013, Valencia, Spain. New York: ACM Press, 2013: 1-16.
- [3] BERNSTEIN D. Containers and cloud: from LXC to Docker to Kubernetes[J]. IEEE Cloud Computing, 2014, 1(3): 81-84.
- [4] MERKEL D. Docker: lightweight Linux containers for consistent development and deployment[J]. Linux Journal, 2014(239).
- [5] 程梦瑶,王璞:定义下一代 DCOS[J]. 软件和集成电路, 2016(4): 84-86.
- CHENG M Y, WANG P: define the next-generation DCOS[J]. Software and Integrated Circuit, 2016(4): 84-86.
- [6] 张基恒,魏进武. 电信运营商私有云适应性分析 [J]. 移动通信, 2015(13): 5-11.
- ZHANG J H, WEI J W. Analysis on private cloud adaptability for telecom operators [J]. Mobile Communications, 2015 (13): 5-11.

[作者简介]



张基恒(1990-),女,中国联合网络通信有限公司研究院工程师,主要从事大数据、IT 架构方面的研究工作。



李大中(1976-),男,中国联合网络通信有限公司信息化事业部工程师,主要从事大数据架构规划、数据安全访问控制、大数据能力开放的设计和研究工作。



张呈宇(1988-),男,中国联合网络通信有限公司研究院工程师,主要从事虚拟化技术、云计算 IaaS、容器技术、数据中心操作系统方面的研究工作。



魏进武(1978-),男,博士,中国联合网络通信有限公司研究院软件与系统实验室主任、副教授,负责和参与了国家“863”计划重大项目 5 项、国家科技重大专项课题 2 项,负责中国联通研发项目 50 余项,主要从事大数据、云计算以及电信 IT 系统等的设计及研发工作。