

周相兵 

云计算软件工程：云软件自动生成原理及方法

Cloud Computing Software Engineering:

Theory and Approach to

Cloud Software Automatic Generation

云计算软件工程：云软件自动生成原理及方法

周相兵 著



科学出版社

北京

内 容 简 介

本书重点介绍了云软件生成的相关理论和方法以及前沿成果,集中反映了作者近年来对云计算与软件工程相结合的研究成果,系统阐述了云计算软件工程的基本结构、技术要领和应用方法。全书共分10章,包括云计算服务模型,服务与云计算研究进展,云软件生成体系结构,虚拟资源粒子群优化分配,云存储遗传优化方法,虚拟资源管理,云软件生成策略,云服务质量,寻址中断的云软件生成方法和云软件部署及案例等。本书大部分云计算软件工程的相关理论和方法均为作者的研究成果。

本书可以作为高等院校计算机、信息管理及相关专业研究生的指导用书,也可供企事业单位、科研机构等部门从事云计算的有关人员和教师参考。

图书在版编目(CIP)数据

云计算软件工程:云软件自动生成原理及方法 / 周相兵著. —北京:科学出版社, 2014.11

ISBN 978-7-03-042327-6

I. ①云… II. ①周… III. ①计算机网络-软件工程 IV. ①TP311.5

中国版本图书馆 CIP 数据核字 (2014) 第 251015 号

责任编辑:李小锐 杨 岭 / 责任校对:韩雨舟

责任印制:余少力 / 封面设计:墨创文化

科学出版社出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

成都创新包装印刷厂印刷

科学出版社发行 各地新华书店经销

*

2014年11月第 一 版 开本:720×1000 B5

2014年11月第一次印刷 印张:14

印数:1—1500册 字数:280千字

定价:64.00元

前 言

软件系统研发大体经历了面向过程、面向对象、面向构件、面向服务以及面向云计算(云服务)、面向大数据的发展历程。目前软件研发主流仍是以面向对象和面向服务为基础,以面向云服务为重要发展方向。就云计算软件工程而言,面向云服务的软件应具有强伸缩性、超弹性以及高扩展性,并且能快速生成、共享和重用 IT 应用程序;在面向用户上,云服务具有无硬件、无维护、无升级等多种优势;而云计算软件工程应具备的特征是软件基础构架稳定、资源 Internet 化、按需在线演化、业务构件服务化、按业务逻辑获取数据自动化,并以潜在数据分析处理促进软件升级,满足服务质量、成熟度和 SLA 要求的软件健壮性、敏捷性、方面性、可用性和扩展性。因此,本书在云计算环境下,以云计算软件工程为引导,围绕云软件自动生成展开研究,即研究了在整个云软件生成的过程中所满足的体系结构、虚拟资源优化分配、云存储优化存储以及云软件生成时的虚拟资源统一管理;同时,为更好地且生成有效的云软件,还研究了云软件生成的基本策略,云软件生成时云服务质量方法以及寻址中断方法。所谓云软件就是在云计算环境下,生成满足不同用户需求的软件系统,而且该生成的软件系统不仅具有云服务化及云计算的特征,还应具有很好的可用性。

全书共有 10 章。

第 1 章,云计算服务模型。主要描述了 SaaS、PaaS、IaaS、DaaS 云服务模型和目前主流开源软件对比,并且也描述了 Hadoop 和最新 YARN。

第 2 章,服务与云计算研究进展。主要综述了面向 QoS 的服务计算方法,面向虚拟化和资源分配的云计算进展以及以 SLA 保障的 QoS 云服务及相关方法。

第 3 章,云软件生成体系结构。在云计算环境下构建了面向云服务的云软件生成体系结构,并给出了云服务应用接口和定义方法。

第 4 章,虚拟资源粒子群优化分配。描述云资源表达方法,并以此提出了基于粒子群的云资源优化分配方法,有效提高了云资源分配的效率。

第 5 章,云存储遗传优化方法。根据云存储结构,提出了一种基于遗传算法的云存储优化方法,大大提升了存储的利用率。

第 6 章,虚拟资源管理。描述了云资源迁移、调度、能量控制和情感负载均衡算法,有效提升了虚拟资源管理效率。

第 7 章,云软件生成策略。描述了云服务语义化描述方法,提出了云服务访

问控制方法：SLA-R，描述了云软件生成策略、流程和方法。

第8章，云服务质量。针对云软件生成过程的云质量，提出了云服务质量指标和方法，并采用遗传算法实现了云服务质量权重树的生成，有效约束了云软件生成的稳定性和收敛性。

第9章，寻址中断的云软件生成方法。提出了一种云软件生成的寻址中断方法和云软件生成的策略、流程。

第10章，云软件部署及案例。描述了云软件部署的指标体系，并以一个智慧旅游平台的导游软件生成为例进行了说明和分析。

特别感谢马洪江教授、苗放教授和余堃教授的指导、支持和帮助。他们对本书倾注了大量心血，特别在研究过程中，给予了全程式关怀、指导和鼓励，促使我走向更宽广的研究领域。

感谢凌道明、杨小平、吴天德、符红霞、马昌威、陈林、李清茂、杨兴江、向昌成、黄成兵、杨文晖等给予的帮助、支持和建议。还要感谢徐海、谢成锦、张敏、苏林梅、刘全飞、廖山清、李睿等给予的帮助。正是有了来自各方的支持、指导、帮助、建议和鼓励，本书才得以顺利出版。

在本书研究的过程中，得到了四川省应用基础研究项目“大数据模式下一种面向智慧旅游的云服务推荐系统研究”（2014JY0005）、四川省科技支撑项目“基于云计算及位置计算技术的智慧旅游信息服务平台应用示范”（2014GZ0013）、四川省哲学社会科学重点研究基地——区域公共管理信息化研究中心项目“云计算视角下的智慧旅游应用模式与策略研究”（QGXH 13-07）、四川省哲学社会科学重点研究基地——四川旅游发展研究中心项目“智慧旅游的前景、挑战和实施策略（LYC13-13）”等科研项目资助和阿坝师范高等专科学校学术专著出版基金资助。

在本书撰写过程中，虽参考了大量中外文献资料，但由于未能查阅更为广泛的文献和篇幅原因，可能相关文献未能一一列出，在此对所有参考文献作者表示诚挚的谢意。

本书在研究和撰写过程中，虽几经努力，但限于作者能力和水平，难免有不足之处。因此，恳求各位专家、学者和广大读者批评指正。宝贵的建议或意见可通过邮箱 3dsmaxmaya@163.com 反馈，谢谢！

周相兵

2014年6月

目 录

第 0 章 导言	1
第 1 章 云计算服务模型	5
1.1 SaaS 及其分析	6
1.2 PaaS 及其对比	7
1.3 IaaS 及其区分	10
1.4 DaaS 及其区别	13
1.5 HDFS 及 MapReduce	16
参考文献	22
第 2 章 服务与云计算研究进展	25
2.1 面向 QoS 的服务计算研究综述	25
2.2 面向资源分配 SLA 保障 QoS 的云计算研究	33
参考文献	48
第 3 章 云软件生成体系结构	53
3.1 云软件生成体系结构	53
3.2 云软件生成原理及方法	58
参考文献	62
第 4 章 虚拟资源粒子群优化分配	63
4.1 虚拟化资源描述方法	64
4.2 虚拟资源优化分配方法	67
4.3 实验分析	72
参考文献	76
第 5 章 云存储遗传优化方法	77
5.1 云存储结构与特征	78
5.2 云存储的扩散划分方法	80
5.3 基于遗传算法的云存储优化	82
5.4 实验分析	83
参考文献	87
第 6 章 虚拟资源管理	88
6.1 云资源管理方法	91

6.2	情感蚁群负载均衡方法	100
6.3	实验分析	110
	参考文献	118
第 7 章	云软件生成描述	120
7.1	面向云服务的云软件生成策略	120
7.2	云服务访问控制方法	122
7.3	云软件生成操作	129
	参考文献	136
第 8 章	云服务质量	138
8.1	云服务质量描述	138
8.2	云服务模型及遗传优化	147
8.3	实验分析	150
	参考文献	155
第 9 章	寻址中断的云软件生成方法	156
9.1	寻址中断策略	156
9.2	云软件寻址中断生成	166
9.3	实验分析	170
	参考文献	173
第 10 章	云软件部署及案例	174
10.1	云服务部署组成指标	174
10.2	案例设计	181
	参考文献	191
附录 1	云计算常用 PaaS 平台描述	192
附录 2	几种常见的 IaaS 开源管理软件	197
附录 3	目前常用的 NoSQL 数据管理功能概述	204
附录 4	常用的云部署(云编配)自动化工具	215
索引		217

第 0 章 导 言

2006 年 3 月, Amazon(亚马逊)推出了弹性计算云服务。同年 8 月, Google 首席执行官 Eric Emerson Schmidt(埃里克·施密特)在搜索引擎大会(SES San Jose)上首次提出了云计算的概念。此后云计算快速地在工业界和学术界发展起来, 并迅速得到了认可和大力推广。它是继 20 世纪 80 年代大型计算机到客户端/服务器(client/server)的大转变之后的又一次巨变, 也是进一步对浏览器/服务器(browser/server)的一次大扩展和大进步, 使在移动过程中感受各类便捷的应用成为现实, 使处于 Internet 中的任何资源和终端都可以集成在“云”中; 从而使需求用户不再需要了解“云”中基础设施的细节, 不必具有相应的专业知识, 也无须直接进行控制^[1-6], 就可以弹性、伸缩使用基于 Internet 的公有资源, 同时满足安全要求且不易涉及隐私的私有资源。

目前, 云计算的概念没有统一的定义, 但它是一种新兴的商业计算模型, 它是对分布式计算(distributed computing)、并行处理(parallel computing)、网格计算(grid computing)、服务计算(service computing)的进一步扩展, 以构建下一代计算机网络^[6,7]。因此, 不同业界根据自己的要求对云计算给出了自己的定义, 但它们离不开云计算是基于 Internet 的、是弹性可伸缩的, 即云计算描述了一种基于 Internet 的新的信息技术服务增加、使用和交付模式, 通过移动互联网来提供动态易扩展且经常是虚拟化的资源^[8]。

1. 几种常见的云计算定义

迄今为止, 云计算还没有一个统一的定义描述; 站在不同的角度所定义、理解的云计算是不同的, 而且对云计算所表达的范围也有不同的表述, 下面是几种常见的定义。

1) Wikipedia

云计算是一种基于 Internet 的计算方式, 通过这种方式, 共享的软硬件资源和信息可以按需求提供给计算机和其他设备。即云计算是一种通过 Internet 以服务的方式提供动态可伸缩的虚拟化资源的计算模式。

2) IBM

云计算通常简称为“云”, 是一种通过 Internet (含移动 Internet)按需交付计算资源(从应用到数据中心都属于计算资源)和按使用付费的基础架构。并且有如下特征^[9]。

(1)富有弹性的资源：能快速轻松地扩大或缩小规模，以满足用户的需求。

(2)按使用付费：计量服务的使用情况，只需为所用的服务付费。

(3)自助服务：使用自助服务可访问用户需要的所有信息技术资源。

3)NIST(美国国家标准与技术研究院)

云计算是一种模式，把实现随时随地、方便的、按需分配的网络访问赋予一个共享的，可以配置的计算资源池。这些计算资源可以快速分配和释放，而只需极少的工作和服务提供商的互动，从而使得云计算有下列特征^[10]。

(1)按需自助服务，随时随地用任何网络设备访问。即消费者在需要网络资源的时候，不必与服务提供商人员接触，就直接实现单方面、自动地获得计算能力，如服务器时间、网络和存储。同时，用户还可以通过基于网络的标准机制访问计算能力，这些标准机制提倡使用各种异构的胖/瘦客户端(移动终端、平板电脑、笔记本计算机和个人工作站)。

(2)多人共享资源池，实现快速重新部署灵活度，即服务提供商的网络资源使用多租户模式来服务多个消费者，并依据用户的需求，将不同的物理和虚拟资源动态地分配和再分配给需求者。同时它还有位置无关的特性，也就是说用户通常不能掌控或者了解资源的具体物理位置，不过用户可以在更高层次的抽象层指定位置(国家，省或者数据中心)。其典型的资源包括存储、处理、内存和网络带宽。

(3)可评测的服务，即通过利用与服务匹配的抽象层次的计量能力(如存储、处理、带宽和活跃用户账号数)，云系统自动控制和优化资源的使用。资源使用可以被监控、控制和报告，提供透明度给服务提供商和服务使用者。

(4)基于虚拟化技术快速部署资源或获得服务。

(5)减少用户终端的处理负担，降低了用户对于信息技术专业知识的依赖。

(6)实现基于 Internet 的快速弹性能力，即弹性的提供或者释放计算能力，以快速伸缩匹配等量的需求，并且在某些情况下，这种伸缩是自动的。而且对于消费者，这种可分配的计算能力通常是无限的，并且可以在任何时候自助任何数量。

4)国内云计算专家刘鹏^[11]

云计算将计算任务分布在大量计算机构成的资源池上，使各种应用系统能够根据需要获取计算能力、存储空间和各种软件服务。

2. 云计算的多租户(项目)模式

多租户/多租赁 (multi tenancy/tenant)是云计算的主要特征之一，是实现云的可伸缩和弹性的重要途径。在云计算环境中，它是一种软件构架，即在一台服务器上运行单个应用实例和按需的项目，为多个租户提供云服务。并且在这种构架下，应用程序设计成将数据、配置进行虚拟化分区后，能使每个租户根据自己

的需求定制自己私有的软件实例(application instance)，也可以供多个组织使用，实现共享同样的应用程序，即应用程序的虚拟化，从而让多个租户共同使用同一个软件实例，但要确保每一个租户的数据、配置等的安全和隔离。也就是说，多租户模式下需要使每一个租户都有自己的一套完整的应用程序、中间件、操作系统和基础构架等，使其与其他租户完全隔离，但可以将数据共享在同一个物理存储、操作系统、中间件和应用程序上。从而构成一种独特的多租户数据中心(大数据)、服务、事务以及基础构架和应用程序运行模式。其中数据是多租户的核心，所面临的安全、容灾挑战也巨大，因此需要通过专业的供应商提供和维护；服务是在软件应用程序中构建的并直接通过 REST(Representational State Transfer)接口和 WSDL(Web Services Description Language)接口访问服务终端，从而建立一种多租户的和 SOA(Service Oriented Architecture)的应用程序；云事务则用来验证每个应用程序的多租户请求，并实现应用程序中的认证和授权以提高多租户的可伸缩性、可维护性和重用性；而基础构架是多租户应用的支撑和载体，是应用程序运行在 PaaS 和 SaaS 上的基础。

在多租户具体实现方面，目前 Java EE7 上已初步提供了部分多租户开发能力，并在 JSR342、JSR338、JSR340、JSR343 和 JSR345 中已有体现，如表 0-1 所示，即对 PaaS 环境中的同一个 PaaS 能够被多个租户使用，并且每个租户使用不同的软件实例(SaaS)和资源共享。

表 0-1 Java EE7 中涉及多租户的 JSR(Java Specification Request)规范

JSR 名称	规范	多租户支持描述	共性
JSR338	JPA2.1	数据方面	每个租户都有唯一的 tenantId
JSR340	Servlets3.1	安全、资源、会话状态和 Web 应用等方面	
JSR342	Java EE7	PaaS 执行环境方面	
JSR343	Java Message Server2.0	消息服务方面	
JSR345	EJB3.2	改进 EJB(Enterprise Java Bean)构架实现数据方面	

参考文献

- [1] Danielson K. Distinguishing cloud computing from utility computing [EB/OL]. http://www.ebizq.net/blogs/saasweek/2008/03/distinguishing_cloud_computing [2012-06-12].
- [2] Hayes B. Cloud computing[J]. communications of the ACM, 2008, 51(7): 9-10.
- [3] Wang L Z, Von Laszewski G, Younge A, et al. Cloud computing: a perspective study[J]. New Generation Computing, 2010, 28(2): 137-146.
- [4] Marston S, Li Z, Bandyopadhyay S, et al. Cloud computing — the business perspective[J]. Decision Support Systems, 2011, 51(1): 176-189.
- [5] Dikaiaikos M D, Dimitrios Katsaros, Mehra P, et al. Cloud computing: distributed Internet computing for IT and scientific research[J]. IEEE Internet Computing, 2009, 13(5): 10-13.
- [6] Wei Y, Brian Blake M. Service-oriented computing and cloud computing: challenges and opportunities [J]. IEEE Internet Computing, 2010, 14(6): 72-75.
- [7] Rings T, Grabowski J, Schulz S. Grid and cloud computing: opportunities for integration with the next generation network[J]. Journal of Grid Computing, 2009, 7(3): 375-393.
- [8] Gruman G. What cloud computing really means [EB/OL]. <http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031> [2012-06-13].
- [9] IBM. 明白了解云计算——在云采用方面实现业务成功的关键在于健壮，成熟的架构 [EB/OL]. http://www-31.ibm.com/ibm/cn/cloud/pdf/Understanding_cloud.pdf [2012-6-13].
- [10] Mell P, Grance T. The NIST definition of cloud computing [EB/OL]. <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf> [2012-06-13].
- [11] 刘鹏. 云计算[M]. 2版. 北京: 电子工业出版社, 2011.

第 1 章 云计算服务模型

从导言的几种云计算的定义和云计算多租户（项目）情况来看，云计算是以 Internet/mobile Internet 为计算基础的一种新型分布式计算模式，因此可通过终端设备、客户端和 Web 浏览器等多种应用模式实现企业更迅速地部署应用程序，并降低管理的复杂度、研发/维护成本和缩短开发周期，以及允许信息技术资源的迅速重新分配以适应企业需求的快速改变^[1-5]。

当前，云计算的服务模式主要以基础设施即服务 (Infrastructure as a Service, IaaS)、平台即服务 (Platform as a Service, PaaS)、软件即服务 (Software as a Service, SaaS) 和数据即服务 (Data as a Service, DaaS) 四种方法来表示云服务和实现对云计算的应用，其中 SaaS 和 PaaS 直接为用户提供服务，IaaS 提供云计算应用的基础资源设施和虚拟化服务，表 1-1 是这四种服务模式的对比。云服务以这四种基本服务模式为基础，通过 Web Service (WSDL)/SOAP (Simple Object Access Protocol) 和 REST (Representational State Transfer) 以及支持数据处理、操作的接口实现 Internet 中的资源搜索、访问和存储，最终生成满足定制和需求的应用程序^[3,6]，满足伸缩和弹性计算要求，以最大化利用网络中的资源。下面对四种基本的云计算服务模式描述如下。

表 1-1 IaaS、PaaS、SaaS 和 DaaS 应用对比

名称	模式	特征	优势	表现形式	共性
IaaS	集成	独立于基础架构平台、实现资源高共享、降低成本、满足服务水平协议 (SLA)、实现按需付费、具备自伸缩等	避免硬件和人力资源上的资本投入、ROI (Return on Investment) 风险减小、简化和自动化伸缩	支持网格计算、服务计算、资源虚拟化、可租用	虚拟化、可租用、安全性、负载均衡、资源共享、计算监控
PaaS	部署	云基础架构、服务敏捷项目管理方法	简化了的版本部署	解决方案堆栈	
SaaS	定制	服务水平协议、瘦客户端应用程序驱动的 UI (User Interface)、云组件、通过 SOAP、REST 通信、无状态、松散耦合、模块化、语义互操作性	无升级无维护无硬件、避免软件和开发资源上的资本投入、ROI 风险减小、简化和迭代的更新	瘦客户端、C/S 应用程序、移动设备应用程序	
DaaS	运营	以数据为中心为不同需求的租户/客户提供面向公共数据 (云数据) 的服务	可以从海量的数据中获取有用的数据，即在未明确定义和非结构化的数据生态中通过非关系数据存储、处理和分析模式挖掘到有用且实时有效的数据	基于云计算进行数据分析处理，即正确使用大数据	历史数据

1.1 SaaS 及其分析

SaaS 是一种软件交付的租赁模式(delivery model)，使相关的数据在云端集中式地托管(hosted)，并以虚拟化的方法为用户提供服务，同时 SaaS 要求在一个 PaaS 上开发的应用程序要能在另一个兼容的 PaaS 上工作。当消费者使用应用程序时，并不掌控操作系统、硬件或运作的网络基础架构，如 Microsoft CRM 与 Salesforce.com。根据高德纳集团(Gartner Group)的评估^[7]，SaaS 的销售在 2010 年达到了 100 亿美元，在 2011 年达到了 121 亿美元，比 2010 年上升了 20.7%。高德纳集团估计，到 2015 年，它的收入将会超过 2010 年收入的 2 倍以上，并且达到预计的 213 亿美元。客户关系管理系统持续成为 SaaS 的最大市场。

在具体的应用中，SaaS 就像一个共享同一个应用程序、保持应用程序数据隔离、满足所有租户应用要求、确保每个租户应用安全的软件版本；实现所有租户在 IaaS 和 PaaS 解决方案中共享所有有用的、满足不同需求的资源，并根据云服务提供商将确定的云应用程序版本安装在多台虚拟机或多个机器上，达到支持扩展和伸缩；同时借助 SaaS 共享、隔离和多租户性质顺利添加新租户，生成一个新的云应用程序，满足用户需求定制的 UI 界面，表 1-2 是常见的基于 Java Script 的 UI 应用包对比，从而无须担忧构建新应用程序带来的不确定性和复杂性，就能提高整个应用环境的部署速度、增加用户接受率、减少支持的需要、节约开发时间、降低实现和升级的成本。其中，Y 代表是，N 代表否。但是开发具有高度伸缩性的 SaaS 应用程序和节约整个 SaaS 软件生命周期是以牺牲安全性和客户隔离需求为代价和以高速有效的移动互联网(含互联网)为基础的。

表 1-2 常见的基于 JavaScript 的 UI 应用包对比^[8]

对比项	Prototype	jQuery	YUI	ExtJS	MooTools
基本描述	不提供开箱即用 UI 组件和 UX 增强功能的 JavaScript 框架之一	jQuery 核心库中包含一些基本的 UX 增强	(Yahoo! User Interface (YUI) 库包含一个 bucketload)	和 YUI 一样，ExtJS 包含大量 UI 组件	与 jQuery 一样，它包含一组有限的效果，其更高级的 UX 增强包含在 MooTools. More.js 扩展中
许可证	MIT	MIT &. GPL	BSD	Commercial &. GPL v3	MIT
支持 IE	6.0 及后续版本	6.0 及后续版本	6.0 及后续版本	6.0 及后续版本	6.0 及后续版本
支持 Firefox	1.5 及后续版本	2.0 及后续版本	3.0 及后续版本	1.5 及后续版本	2.0 及后续版本
支持 Ajax	Y	Y	Y	Y	Y
DOM 操作	Y	Y	Y	Y	Y
DOM 遍历	Y	Y	Y	Y	Y
事件处理	Y	Y	Y	Y	Y
JSON	Y	Y	Y	Y	Y
选择器	Y	Y	Y	Y	Y
Accordion	N	jQuery UI	N	Y	MooTools More

续表

对比项	Prototype	jQuery	YUI	ExtJS	MooTools
动画	scriptaculous	Y	Y	Y	Y
自动完成	scriptaculous	N	Y	N	N
浏览器历史	scriptaculous	N	Y	Y	N
日历	N	jQuery UI	Y	Y	N
绘图	N	N	Y	Y	N
拖拽	scriptaculous	jQuery UI	Y	Y	MooTools More
网格	N	N	Y	Y	MooTools More
进度条	N	jQuery UI	Y	Y	N
重新调整大小	N	jQuery UI	Y	Y	N
富文本编辑器	N	N	Y	Y	N
滑块	scriptaculous	jQuery UI	Y	Y	MooTools More
选项卡	N	jQuery UI	Y	Y	N
主题	N	jQuery UI	Y	Y	MooTools More
树视图	N	N	Y	Y	N

1.2 PaaS 及其对比

PaaS 是在 IaaS 和 SaaS 间的一种终端运算服务，也就是说 PaaS 是 IaaS 之上的一层，是基于 IaaS 存在的；同时也要满足一个 IaaS 上工作的平台要与另一个 IaaS 上工作的平台相兼容，它提供了 SaaS 需要的中间组件、数据库、存储、连接、可靠性、缓存、监视和路由。这样就可以简化云应用程序开发，实现简单、灵活的资源分配以及各种工具和服务。PaaS 应用场景之一就是在 IaaS 上向用户提供一种按需支撑运行的应用运行平台（Application Running Platform，ARP），即 ARP 在需求与运行策略库、规则库、软件映像库、资源库（池）、平台交付组件和运行时控制组件下向 SaaS 提供基于 WSDL/REST 应用需求接口，从而使满足应用需求的云软件具备快速应用部署环境运行能力，实时使基于应用需求快速构建应用运行环境的能力、运行实时动态满足应用需求的能力、支持多租户运行的能力、满足安全运行能力、数据共享与隔离的能力等。

目前支持 PaaS 的语言主要有 Manjrasoft Aneka，Amazon Elastic Beanstalk（AEB）^[12]、CloudBees、Cloud Foundry（CF）、Google App Engine（GAE）、Red Hat OpenShift（RHO）、Jelastic、HeroKu、Microsoft Windows Azure（MWA）、Force.com、Engine Yard Cloud（EYC）、CumuloLogic、Engine Yard Orchestra（EYO）等，表 1-3 展示了目前支持 PaaS 的主要语言对比。而常见的 PaaS 平台主要包括 GAE^[9]、MWA^[10]、Force.com^[11]、Amazon Web Services（AWS）^[13,14]、VMware Cloud Foundry（CF）、RHO、IBM SmartCloud Application Services（IBM SCS）^[15]。其中，Y 代表是，N 代表否。详细介绍参见附录 1，其中附表 1-1 介绍几种常见 PaaS 平台，附表 1-2 是 AWS 中的 EC2、S3、EBS、SQS。

表 1-3 目前支持 PaaS 的主要语言对比表

名称	Aneka ^[16]	AEB ^[12]	CF ^[17]	GAE ^[9]	Jelastic ^[18]	RHO ^[19]	CloudBees ^[20]
支持语言	.NET	Java/.NET/PHP	Java/Spring/ Groovy/net/Node.js/Grails/Ruby Rails & Sinatra	Java/ Python/PHP	Java/PHP/Ruby/ Scala/Groovy	Java/Java EE/Python/Perl/PHP/ Ruby/Node.js	Java
开源状况	N	N	Y	Medium	N	Y	N
存储装置	SQL Server/S3	S3	und	Storage/Blobstore	Maven/Git/SVN	Git/SSH/Rsync/ Maven/SFTP	Maven/Git/SVN
数据库	SQL Server SimpleDB MySQL	Relational Database Service/ SimpleDB/PostgreSQL	MongoDB/ MySQL/Redis	Bigtable	Maria DB/MySQL/ PostgreSQL/Mon-goDB/CouchDB	MySQL/PostgreSQL/Mon-goDB/ MySQL	MySQL
特征描述	它是一种面向市场的云计算平台，它允许用户构建和调度应用程序，并提供资源，并在私有云或公有云环境中使用定价、核算与 QoS/SLA 的服务监测结果；且可与 EC2 构建混合云；全面支持 MapReduce	内置 PHP 和 Tomcat 的 HTTP 软件栈；能将 WAR 文件部署到一个无固定形状的、极度可扩展的环境中；提供 Tomcat 实例；与 REST API 集成可访问关系数据库；简化了 Tomcat 应用程序的部署和扩展	支持创建一个能够运行 Node.js 的私有 PaaS；具有很好的可移植性且能运行在 Amazon EC2 中；由 VMware 发布	提供了 IDO/JPA 持久化数据的能力；扩展 Grails；线性扩展能力强；支持大规模操作访问；限制 Java 程序，如不支持 Spring、Struts 等	对纯 Java 有较好的支持；支持 Java 流行应用程序容器；使用基于容器的虚拟化方案；通过宿主主易得到 J2EE 容器；通过采用独特的自动垂直扩展能力实现 CPU、RAM 动态分配	OpenShift 的基础是 RHEL；利用 RHEL 操作系统自身的控制组和 SELinux 实现隔离的安全机制；用 RHEL 的资源管理实现多租户环境设置；用 RHEV 实现虚拟化用 JBoss；中间件实现 J2EE 全支持	对纯 Java 有较好的支持；支持 Java 流行应用程序容器；旨在在受管理的伸缩性和灵活性之间发现正确的平衡；目前基于 EC2 的基础设施；提供可扩展的基础设施；托管的 MySQL 关系数据库；能与 S3、SQS 和 SES 服务集成；是 AEB/RDS 低成本替代品

续表

名称	HeroKu ^[21]	MWA ^[10]	Force.com ^[11]	EYC ^[22]	Cumulogic ^[23]	EYO ^[24]
支持语言	Java/Ruby/Clojure/ Node.js/Python/Scala	· N/PHP/Node.js/Py- thon	Apex/Visualforce	Node.js/PHP/Ruby	Java/Spring, PHP/Py- thon/ Ruby	PHP/Ruby/Node.js
开源状况	N	N	N	Medium	Medium	Medium
装置载体	Git	BLOB (Binary Object)	Force.com IDE	Git/Oracle Virtual Box	Eclipse/NetBeans	Git/Subversion
数据库	RDS/MySQL/Mon- goDB/CouchDB/Re- dis	SQL Server		MySQL/RDS/SQL Server/PostgreSQL	MySQL/MongoDB/ CouchBase/Percona XtraDB/RDS	MySQL/RDS/ SQL Server
特征描述	将 Ruby 的灵活、乐 趣第一的应用程序开 发方法与 Git 的智能 分布式部署模型结合 了；提供了比 GAE 更多的控制权，但它 提供的控制权少于 AEB；它的命令行工 具是用 Ruby 编写的； Heroku 与 Git 的集成 能提供一种通过云部 署和扩展 Java 应用程 序的新模式	允许开发人员在微软管 理的数据中心内跨网进 行应用程序部署；通过 人工角色托管 Tomcat 服务器和 JVM，而 Az- ure SDK 则可以支持 Java 开发； Azure HPC Scheduler 可帮助人们 简化计算密集型并行应 用程序的开发；提供 SQL Azure 数据库服务 和操作；提供 Windows 和 Linux 虚拟机；功能 和服务采用开放式 REST 协议；提供了类 似 Salesforce 的应用级 的服务	允许开发者构建具有社 交和移动特性的应用程 序；提供了有助于在云 上更快建立和运行业务 应用程序的所有功能； 开发者可以基于 UI 层 面编写数据库触发器和 程序控制器；提供了冗 余、安全和可扩展等常 见特性；自身提供了托 管服务，并且更加侧重 于数据处理而不是代码 处理；提供了丰富的外 部编程访问；使用了一 个外部搜索引擎，提供 完整的索引并允许搜索 非结构化数据	提供一个 Rubyon Rails 的技术栈；目 前提供一个 Rails 优 化的 Linux 发行版； 开发者可以通过 Engine Yard 的仪表 板进行环境配置、 应用程序部署和 监控	提供了支持将应用迁移 到云端的平台；可以将 虚拟化环境或者 IaaS 传 输到应用平台上，并对 它们进行实时监控、自 动故障修复和管理；部 署和管理应用提供了管 理应用层，从而帮助开 发人员实行更好的协作； 使开发者可以开发移动 \ HTML5 \ Web 或者 企业级应用；很好地支 持 AWS(AEB)、 Rabbit- MQ	构建于亚马逊的 AWS 服务平台上，也能基于 Windows Azure 提供云 服务；应用程序可以与 内置的内容管理系统进 行部署；应用程序还可 以根据使用情况 and 流量 自动扩展；对于运行的 API、Web 服务和移动 应用也具有出色表现； 通过建立的 Flex 为客户 提供不间断的云服务； 可以部署高扩展性、高 并发的应用程序

1.3 IaaS 及其区分

IaaS 是云计算的基础设施，是实现 PaaS 和 SaaS 的保障，它处于云计算的最底层，使多租户能共享如 CPU、网络、存储和操作系统等基础构架资源，为用户提供计算容量、可通过网络访问的存储和一个镜像操作系统。在提供 IaaS 服务的云计算中，若需要，每个租户都可以有效地得到一个虚拟机，而这些虚拟机以密集的方式运行在后台服务器集群中；虚拟机的特点就是提供给租户类似于物理机的体验；换句话说，IaaS 是消费者使用处理、储存、网络以及各种基础运算资源，部署与执行操作系统或应用程式等各种软件；多租户客户端无须购买服务器、软件等网络设备，即可任意部署和运行处理、存储、网络和其他基本的计算资源，不能控管或控制底层的基础设施，但是可以控制操作系统、存储装置、已部署的应用程序，有时也可以有限度地控制特定的网络组件，如主机端防火墙^[25,26]。在具体应用中，在一个 IaaS 上工作的平台应与另一个 IaaS 上工作的平台相兼容。

表 1-4 是目前最常用的 IaaS 管理的开源软件(相关开源软件的描述见文献[27])，能最大限度提供灵活的云计算和存储基础构架管理，而且包括了 IaaS 所需的虚拟化、接口和安全性等必要功能。能管理各种故障的虚拟机而不会影响到其上运行的业务和服务/存储/网络层的整体计算能力，支持主流操作系统平台镜像的创建、捕获、管理、部署，能自我管理并故障修复，能有效提供命令行工具、Web 服务应用程序接口(Application Program Interface, API)和基于 Web 的 UI，能为用户提供多种渠道申请、使用、管理自己在云中的资源，集成监控与统计信息等功能，从而实现云计算的数据中心要求的高可用性、安全性和可扩展性，精简、集中和存储资源的虚拟化，弹性伸缩，适应不断变化的业务需求、私有或混合云环境的选择灵活性，企业级工作负载监控服务、内置的资源可用性分析和支持决策等^[28]。表 1-4 是 IaaS 主流 OpenStack^[29,30]、CloudStack^[31]、Eucalyptus^[32,33] 与 OpenNebula^[34-36] 的对比，常见的 IaaS 平台还有 Nimbus^[37]、ConVirt^[36,38]、OpenQRM^[36,39]，图 1-1、图 1-2 是截至 2013 年 9 月这四款 IaaS 开源软件社区活跃变化趋势图^①。几种常见的 IaaS 开源管理软件参见附录 2。

表 1-4 四种主流 IaaS 开源软件对比

对比项	CloudStack	OpenStack	Eucalyptus	OpenNebula
云服务模式	IaaS	IaaS	IaaS	IaaS
所处云计算层次	底层	底层	底层	底层
授权协议	Apache2.0	Apache2.0	GPLv3	Apache2.0

① <http://WWW.qyjohn.net/?P=3399>.