

“西安石油大学优秀学术著作出版基金”资助

多租户

关键技术

朱养鹏 著

中国石化出版社

[HTTP://WWW.SINOPEC-PRESS.COM](http://www.sinopec-press.com)

多租户关键技术

朱养鹏 著

中国石化出版社

图书在版编目(CIP)数据

多租户关键技术/朱养鹏著. —北京: 中国石化出版社, 2014. 10
ISBN 978-7-5114-3057-1

I. ①多… II. ①朱… III. ①云计算-研究 IV. ①TP393.027

中国版本图书馆 CIP 数据核字(2014)第 228101 号

未经本社书面授权, 本书任何部分不得被复制、抄袭, 或者以任何形式或任何方式传播。版权所有, 侵权必究。

中国石化出版社出版发行

地址: 北京市东城区安定门外大街 58 号

邮编: 100011 电话: (010)84271850

读者服务部电话: (010)84289974

<http://www.sinopec-press.com>

E-mail: press@sinopec.com

北京富泰印刷有限责任公司印刷

全国各地新华书店经销

*

710×1000 毫米 16 开本 9 印张 153 千字

2014 年 10 月第 1 版 2014 年 10 月第 1 次印刷

定价: 45.00 元

前言

云计算是近年来兴起的具有深远影响的 IT 技术,是分布式计算、并行计算、效用计算、网络存储、虚拟化等技术发展融合的产物,得到广泛的研究与应用。多租户技术是云计算的重要部分,具有按需租用,升级维护方便、为租户和软件供应商节约成本、便于系统扩展等优点。

通过多租户技术,可用单个应用实例为多个租户提供计算、存储以及应用等多种服务,使多个租户共享软硬件资源,提高资源利用率,降低分摊在单个租户上的平均基础设施和管理成本。

但是,随着云计算环境下的多租户技术应用日益增多,一些问题也不断浮现出来,例如:在多租户共享软硬件资源的情况下,如何保证各租户之间的性能隔离;在各租户的 SLA (Service Level Agreement, 服务等级协议) 各不相同,如何保证多个租户的服务质量同时得到满足;多租户环境下隔离机制以及实现算法;多租户环境下的访问控制等。

针对上述问题,本文深入研究了云计算平台环境下的多租户关键技术。主要研究工作与创新点如下:

(1) 提出了一个基于虚拟化的多租户应用管理平台

多租户应用管理平台由应用管理层、资源管理层和基础设施层组成。该平台以虚拟化技术为基础,通过虚拟机进行租户的逻辑隔离,通过租户资源分配模型建立租户 SLA 要求同租户资源之间的关系,并通过租户安置算法将租户安置在合适的服务器上;通过租户资源动态管理框架根据租户的实时负载情况调整租户的资源数量,在满足租户 SLA 要求的基础上,最大限度的提高资源利用率;通过多租户访问控制模型加强了在多租户环境下用户的访问安全问题,并提出了多租户跨安全域访问方法;通过应用迁移模块实现了现有应用向多租户模式迁移的方法,降低了多租户应用开发的复杂度。

(2) 提出了基于租户 SLA 的资源消耗模型以及租户安置算法

在租户安置阶段,为了最大限度提高平台资源利用率,确保租户服务质量,研究了租户 SLA 要求和所需资源之间的关系,定义了租户资源消耗模型,该模型通过租户 SLA 要求可以计算出租户所需的资源数量。通过租户资源消耗模型将租户 SLA 映射到不同的虚拟机上,将对租户的资源管理转化为对虚拟机的资源管理。并从资源利用的角度出发,提出了基于租户服务质量要求约

束的租户安置算法,该安置算法以遗传算法为基础,采用基于群体的编码方式,为不同长度的染色体设计了交叉和变异遗传算子。该算法对解空间进行全局寻优搜索,具有自组织、自适应和自学习性,可以迅速找到租户安置的最优解。实验结果证明,与传统的启发式算法相比,本文提出租户资源分配模型和安置算法在保障租户服务质量要求的基础上,能有效减少物理服务器的使用数量。

(3)提出了基于实时负载的多租户资源动态管理框架

在多租户应用的运营阶段,租户的负载会出现周期性或突发性的负载高峰或低谷情况。如果按照负载的峰值情况为租户分配资源,可以保证租户的服务质量,但是会导致租户资源利用率较低;如果按照平均负载为租户分配资源,在租户的峰值情况下会出现租户服务质量不能得到保障。

针对以上情况,为保证租户服务质量并提高资源利用率,本文提出了基于实时负载的多租户资源动态管理框架,该框架首先通过对租户的历史和实时负载进行记录,通过时间序列模型预测租户下一时刻的负载情况,然后租户选择算法根据租户负载预测情况将租户进行分组,负载互补的租户分到相同的组中;最后计算每组租户所需的最少资源总量,并为每组租户分配资源。通过为负载互补的租户共同分配最少的资源总量的方法而不是为每一租户单独分配资源,提高了服务器资源利用率,降低了租户违反 SLA 要求的次数。

(4)提出了多租户环境下的访问控制模型

针对多租户角色复杂、各租户之间数据共存而又独立访问的要求,结合基于角色的访问控制模型,构建了支持多租户、多角色、方便租户权限管理的多租户访问控制模型。该模型在基于角色的访问控制模型的基础上增加了租户的概念,通过租户将各用户的访问和数据从逻辑上分离,提高了多租户平台访问控制的安全性和可管理性。对于多租户应用和企业异构系统相互访问需求,提出了多租户跨安全域访问模型,该模型分为安全访问管理中心和本地访问管理两部分,通过将应用划分为不同的安全域,安全访问管理中心管理不同安全域之间的应用互访的身份认证,本地访问管理域内应用程序之间的安全互访。在技术实现方面,通过在 SOAP 头部加入访问者身份信息扩展 Web 服务的方式实现应用之间的身份传递,保障了多租户和企业异构系统的安全交互。

(5)提出了应用到多租户的迁移技术

针对应用开发商准备将已开发好的应用程序转化为多租户应用的需求,提出了应用到多租户的迁移平台,该平台包括数据库转换模块、租户认证模块、原 Web 系统以及平台管理模块。数据库转换模块负责将原 Web 系统的单租户数据库转换为多租户数据库,并且根据用户所属的租户过滤用户对数据的操

作，保证授权用户只能访问所属租户的数据。平台管理模块负责租户的管理、租费的管理、资源监控、资源调度等功能。使用多租户迁移平台，只需经过少量的代码修改以及简单的应用配置，可以将传统的 Web 应用系统转化为支持云的多租户应用系统，减少了应用开发商开发多租户应用的风险和成本。

在上述研究基础上，以云制造平台为背景，实现了云计算平台环境下的多租户应用管理平台，该平台由租户安置模块、租户资源动态管理模块、多租户访问控制模块以及应用迁移模块组成。并以物料管理系统为例进行了多租户应用的部署、租户的安置以及租户资源的动态管理等方面的实现。

本专著涉及云计算、服务计算两大研究领域，资源丰富、结构新颖、条理清晰、重点突出，可供高等院校计算机及信息管理相关专业研究生、本科生选用，也可以作为云计算应用方面的科技工作者选用。

第1章 绪论	1
1.1 研究背景	1
1.2 研究意义	2
1.3 研究内容	3
1.4 本书结构	4
第2章 云计算技术	5
2.1 云计算概述	5
2.2 云计算模型	6
2.3 云计算部署模式	7
2.4 云计算关键技术	8
2.5 云计算应用	9
2.6 云计算安全	10
2.7 云计算未来	11
第3章 服务计算技术	12
3.1 服务计算概述	12
3.2 服务计算模型	13
3.3 服务计算部署模式	14
3.4 服务计算关键技术	15
3.5 服务计算应用	16
3.6 服务计算安全	17
3.7 服务计算未来	18
第4章 多租户应用管理平台	19
4.1 多租户应用管理平台概述	19
4.2 多租户应用管理平台架构	20
4.3 多租户应用管理平台功能	21
4.4 多租户应用管理平台实现	22
4.5 多租户应用管理平台应用	23
4.6 多租户应用管理平台安全	24
4.7 多租户应用管理平台未来	25
第5章 物料管理系统	26
5.1 物料管理系统概述	26
5.2 物料管理系统架构	27
5.3 物料管理系统功能	28
5.4 物料管理系统实现	29
5.5 物料管理系统应用	30
5.6 物料管理系统安全	31
5.7 物料管理系统未来	32
第6章 结论	33
6.1 研究内容	33
6.2 研究意义	34
6.3 研究内容	35
6.4 本书结构	36
6.5 本书结构	37
6.6 本书结构	38
6.7 本书结构	39
6.8 本书结构	40
6.9 本书结构	41
6.10 本书结构	42
6.11 本书结构	43
6.12 本书结构	44
6.13 本书结构	45
6.14 本书结构	46
6.15 本书结构	47
6.16 本书结构	48
6.17 本书结构	49
6.18 本书结构	50
6.19 本书结构	51
6.20 本书结构	52
6.21 本书结构	53
6.22 本书结构	54
6.23 本书结构	55
6.24 本书结构	56
6.25 本书结构	57
6.26 本书结构	58
6.27 本书结构	59
6.28 本书结构	60
6.29 本书结构	61
6.30 本书结构	62
6.31 本书结构	63
6.32 本书结构	64
6.33 本书结构	65
6.34 本书结构	66
6.35 本书结构	67
6.36 本书结构	68
6.37 本书结构	69
6.38 本书结构	70
6.39 本书结构	71
6.40 本书结构	72
6.41 本书结构	73
6.42 本书结构	74
6.43 本书结构	75
6.44 本书结构	76
6.45 本书结构	77
6.46 本书结构	78
6.47 本书结构	79
6.48 本书结构	80
6.49 本书结构	81
6.50 本书结构	82
6.51 本书结构	83
6.52 本书结构	84
6.53 本书结构	85
6.54 本书结构	86
6.55 本书结构	87
6.56 本书结构	88
6.57 本书结构	89
6.58 本书结构	90
6.59 本书结构	91
6.60 本书结构	92
6.61 本书结构	93
6.62 本书结构	94
6.63 本书结构	95
6.64 本书结构	96
6.65 本书结构	97
6.66 本书结构	98
6.67 本书结构	99
6.68 本书结构	100



目录

1 云计算技术	(1)
1.1 云计算的定义	(1)
1.2 云计算特点	(2)
1.3 云计算的架构	(3)
1.3.1 云计算服务模式	(3)
1.3.2 云计算的部署模式	(8)
1.4 云计算关键技术	(10)
1.4.1 数据存储技术	(10)
1.4.2 数据管理技术	(11)
1.4.3 编程模式	(12)
1.5 云计算与其他概念	(13)
1.5.1 云计算与网格计算	(13)
1.5.2 云计算与 SOA	(14)
1.5.3 云计算与超级计算机	(15)
1.6 云计算存在的问题及下一步技术发展趋势	(16)
2 多租户技术概述	(18)
2.1 研究背景	(18)
2.2 多租户技术研究现状	(19)
2.2.1 多租户模型及体系结构	(20)
2.2.2 租户隔离	(26)
2.2.3 租户资源分配和安置策略	(27)
2.2.4 租户资源动态管理	(31)
2.2.5 虚拟化技术	(35)
2.2.6 多租户关键技术研究方向、趋势、存在问题	(36)
2.3 研究内容、创新点与研究意义	(38)
2.3.1 研究内容	(38)
2.3.2 创新点	(40)

2.3.3	研究意义	(42)
2.4	基于虚拟化技术的多租户应用管理平台	(43)
2.4.1	多租户应用管理平台体系结构	(45)
2.4.2	多租户应用管理平台工作流程	(46)
2.4.3	多租户应用管理平台工作原理	(48)
2.4.4	关键技术	(50)
3	多租户资源消耗模型和租户安置方法	(51)
3.1	引言	(51)
3.2	租户安置框架	(52)
3.3	租户资源消耗模型	(53)
3.4	租户安置模型	(54)
3.5	租户安置算法	(55)
3.5.1	编码	(55)
3.5.2	初始染色体种群的生成	(56)
3.5.3	选择、交叉和变异	(57)
3.5.4	算法流程图	(58)
3.5.5	算法伪代码	(59)
3.6	性能测试	(59)
3.7	小结	(63)
4	多租户资源动态管理框架	(64)
4.1	引言	(64)
4.2	多租户资源管理框架	(67)
4.3	满足租户 SLA 的资源模型	(69)
4.4	超级租户资源总量的计算	(71)
4.4.1	负载预测	(72)
4.4.2	预测误差建模	(73)
4.4.3	计算多租户资源	(74)
4.5	租户选择	(74)
4.6	多租户资源动态管理框架测试	(76)
4.6.1	租户整合	(76)
4.6.2	为租户提供资源保障	(78)
4.7	小结	(79)
5	多租户访问控制研究	(80)
5.1	引言	(80)

5.2	多租户访问控制模型	(83)
5.2.1	用户分析	(83)
5.2.2	多租户访问控制形式化描述	(84)
5.2.3	多住户访问控制模型操作流程	(86)
5.2.4	访问控制模型数据库设计	(88)
5.3	多租户应用跨安全域访问	(90)
5.3.1	跨安全域多租户安全架构	(92)
5.3.2	跨安全域工作流程	(93)
5.3.3	跨安全域的实现	(93)
5.4	小结	(98)
6	应用到多租户的迁移技术	(99)
6.1	引言	(99)
6.2	传统软件和 SaaS 系统的比较	(100)
6.3	支持多租户的 SaaS 转换平台	(101)
6.3.1	数据库转换模块	(101)
6.3.2	身份认证模块	(104)
6.3.3	管理配置模块	(105)
6.4	转换例子	(106)
6.4.1	Web 物料管理系统介绍	(106)
6.4.2	转换过程	(106)
6.5	平台测试	(107)
6.5.1	测试环境的建立	(107)
6.5.2	系统功能和数据隔离测试	(108)
6.5.3	转换平台性能测试	(108)
6.6	讨论	(112)
6.6.1	转换平台的优势	(112)
6.6.2	适合转换为多租户的系统	(112)
6.7	本章小结	(113)
7	原型系统实现	(114)
7.1	多租户应用平台设计与实现	(114)
7.1.1	面向 Xen 的开发技术	(114)
7.1.2	多租户平台模块结构	(115)
7.1.3	多租户平台运行评价	(117)
7.2	应用实例	(118)

7.2.1 供应商注册应用	(118)
7.2.2 租户注册应用	(119)
7.3 小结	(122)
8 结论与展望	(123)
8.1 结论	(123)
8.2 展望	(125)
参考文献	(126)

1 云计算技术

计算机软件与硬件技术的飞速发展也导致了计算模型的不断演化。继分布式计算、并行计算、网络计算、效用计算、Web2.0 等计算概念与模型不断被推出以后,计算机工业界与学术界又提出了云计算模型,在某种意义上实现对这些计算概念与模型的泛化与商业化。总体上来讲,云计算通过互联网将超大规模的计算与存储资源整合起来,并以可信服务的形式按需提供给用户。

1.1 云计算的定义

云计算是计算机资源公共化在商业模式上的一个重要革新。通过云计算,用户将从繁重,复杂,易错的计算机资源管理中解放出来,而只需关注业务逻辑。这种业务逻辑与计算资源的分离将大大地降低企业信息化的复杂度。更重要的是,云计算带来的服务整合与按需供给将大大提高当前计算资源的利用率,降低每服务的能耗量,并且有效屏蔽计算资源的出错问题。云计算正处于不断演化的过程中,各界对云计算的定义持有不同看法。

IBM 蓝云定义:

- 一种新的用户体验和业务模式:云计算是一种新出现的计算模式,它是一个 IT 计算资源池,并将应用、数据及其他资源,以服务的形式通过网络提供给最终用户。

- 一种新的 IT 架构管理方法:云计算是一种新的方式来管理大量的虚拟化的 IT 资源,从管理的角度来看云计算,它可以将多个小的资源组装成大的资源池,也可以将大型资源虚拟化成多个小型资源,而最终目的都是提供服务。

微软认为云计算就是通过 Internet 标准和协议以实用工具形式提供的计算功能。

百度百科给出了狭义云和广义云的概念:

- 狭义云计算:提供资源的网络被称为“云”。“云”中的资源在使用者看来是可以无限扩展的,并且可以随时获取,按需使用,随时扩展,按使用付费。这种特性经常被称为像水电一样使用 IT 基础设施。

- 广义云计算:这种服务可以是 IT 和软件、互联网相关的,也可以是任意其他的服务。

Sun 公司认为,云的类型有很多种,而且有很多不同的应用程序可以使用云来构建。由于云计算有助于提高应用程序部署速度,有助于加快创新步伐,因而云计算可能还会出现我们现在无法想像到的形式。作为创造“网络就是计算机”(The Network is the Computer)这一短语的公司,SUN 公司深信云计算就是下一代的网络计算。

维基百科(Wikipedia.org)对云计算的定义也在不断更新,前后版本的差别非常大。最新定义为:云计算是一种动态的易扩展的且通常是通过互联网提供虚拟化的资源计算方式,用户不需要了解云内部的细节,也不必具有云内部的专业知识或直接控制基础设施。云计算包括基础设施即服务(IaaS)、平台即服务(PaaS)和软件即服务(SaaS)以及其他依赖于互联网满足客户计算需求的技术趋势。云计算主要提供通用的通过浏览器访问的在线商业应用、软件和数据存储等服务。

加州大学伯克利分校(University of California at Berkeley)的 Michael Armbrust 等在名为“伯克利云计算白皮书(Above the clouds: a Berkeley view of cloud computing)”中对云计算的定义:云计算包括互联网上各种服务形式的应用以及这些服务所依托数据中心的软硬件设施,这些应用服务一直被称作软件即服务(SaaS),而数据中心的软硬件设施就是所谓的云,云计算就是 SaaS 和效用计算。

为了更好的促进云计算的发展与应用,美国国家标准与技术研究院(NIST)制定了“云计算工作定义”。这份“工作定义”提出了云计算的基本定义和特征、交付模式和部署模式。云计算是一种资源利用模式,它能以简便的途径和以按需的方式通过网络访问可配置的计算资源(网络、服务器、存储、应用、服务等),这些资源可快速部署,并能以最小的管理代价或只需服务提供商开展少量的工作就可实现资源发布。

1.2 云计算特点

①按需自助服务。消费者可对计算资源(如服务器时间和网络存储)进行单边部署以自动化地满足需求,并且无须与服务提供商的人工配合。

②泛在网络连接。云计算资源可以通过网络获取和通过标准机制访问,这些访问机制能够促进用户通过异构的瘦(thin)客户平台或胖(thick)客户平台(手机、笔记本电脑、掌上电脑等)来使用云计算,也包括其他传统的或基于云的服务。

③与地理位置无关的虚拟化的资源“池”。云计算服务商采用多用户模式,根据用户需求动态地分配和再分配物理资源和虚拟资源。用户通常不用知道这些资源具体所在位置,资源包括存储、处理器、内存、网络、虚拟机等。虽

然存在某种程度上的位置无关性,也就是说用户无法控制或根本无法知道所使用资源的确切物理位置,但是原则上可以在较高抽象层面上来指定位置(例如国家、州、省、或者数据中心)。资源的例子包括存储、处理、内存、网络带宽以及虚拟机等。即使是私有的“云”往往也趋向将资源虚拟“池”化来为组织的不同部门提供服务。

④快速灵活地部署资源。云计算供应商可快速灵活地部署云计算资源,快速地放大和缩小。对于用户,云计算资源通常显得是无限的,并可以在任何时间购买任何数量的资源。

⑤服务计费(可测量的服务)。通过对不同类型的服务进行计费,云计算系统能自动控制和优化资源利用情况。可以监测、控制资源利用情况,并形成报告,为云计算提供商和用户就所使用的服务提供透明性。人们可以监视、控制资源使用、并产生报表,报表可以对提供商和用户双方都提供透明。

1.3 云计算的架构

1.3.1 云计算服务模式

云计算有三种服务模式:把云软件作为服务(SaaS)、把云平台作为服务(PaaS)、把云基础设施作为服务(IaaS)。图1-1给出了这三种云服务模式所对应的云提供商。



图 1-1 云计算服务模式

(1) 软件即服务 (SaaS)

在这种交付模式下, 提供商在云计算设施上运行程序, 用户通过各种客户端设备的瘦客户界面(如网页浏览器、基于网页的电子邮件)使用这些应用程序。用户不需管理或控制底层的网络、服务器、操作系统、存储系统、应用程序等, 但可能需要完成一些与用户相关的应用程序参数设置。“把软件作为服务”的特色是根据需要作为服务提供的一整套应用程序。该软件的单个实例运行于云上, 并为多个最终用户或客户机构提供服务。

相关的 SaaS 示例主要包括:

最著名的 SaaS 示例是 Salesforce.com。1999 年, 27 岁的甲骨文(oracle)高级副总裁、俄罗斯裔美国人马克·贝尼奥夫创办了 Salesforce.com 公司, 提出软件即服务(SaaS, Software-as-a-service)的思想, 并运用于客户关系管理服务(CRM)软件。目前已有 72500 家公司采用了 Salesforce 的 CRM。

Oracle 也推出了 SaaS 平台, 其体系结构如图 1-2 所示。

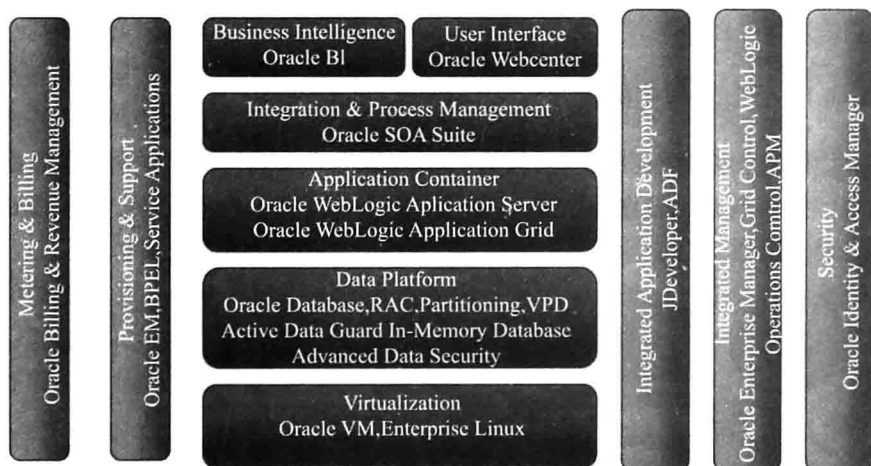


图 1-2 Oracle SaaS 平台架构

针对终端用户谷歌则推出了 Google Apps, 这是一套基于网络的文档、电子数据表和其他生产力的应用软件。主要有 Gmail, GoogleDOCS、Google MAP、Calendar、Chat 等, 如图 1-3 所示。

Microsoft Online Services: 包括 Windows Live、Office Live、Exchange Online、SharePoint Online、Office Live Meeting、Office Communications Online、Exchange Hosted Services、Dynamics CRM Online。

NetSuite: NetSuite 提供的 NetSuite Business Software Suite 是行业领域首家在

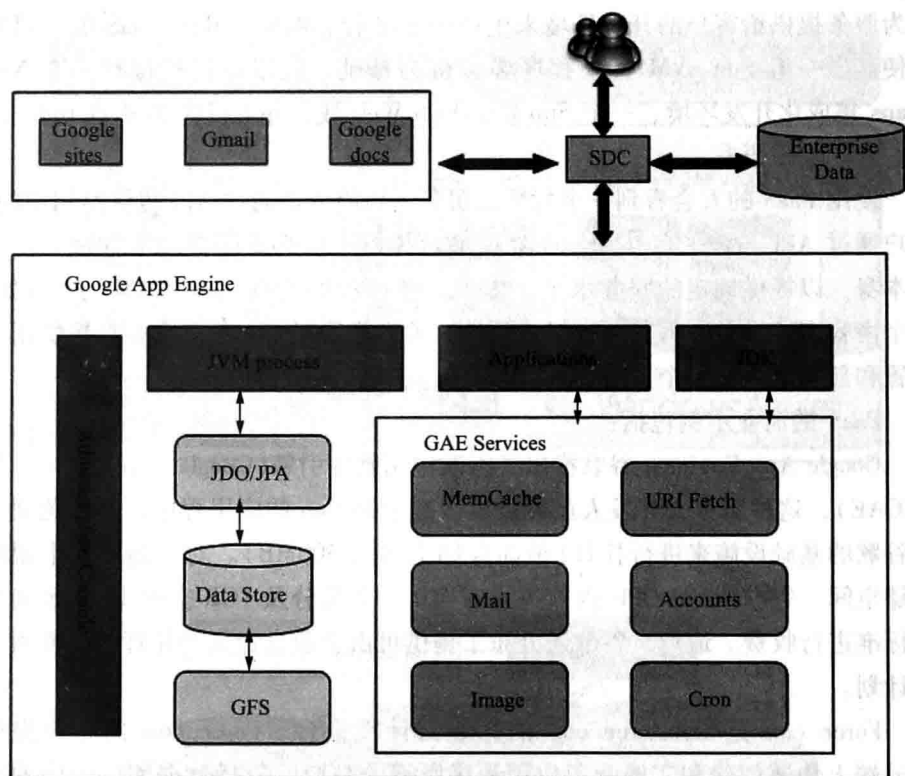


图 1-3 谷歌 SaaS 平台

线商业软件。在强大的单一应用软件中，NetSuite 集成了前台面向最终用户的 CRM 和电子商务功能与后台面向企业内部及合作伙伴的 ERP，以及自助服务门户功能。

Yahoo: Yahoo Small Business 服务提供了一个组合，其中包括：域名注册服务、网站托管服务、邮件服务，这个简单组合在一个服务包内让小企业拥有了所有必要的面向因特网的服务，并且比起其它非云的选择具有明显的价格优势。客户所需的只是接入服务，在不再需要时可以简单地取消服务。

(2) 平台即服务 (PaaS)

在这种交付模式下，用户采用提供商支持的编程语言和工具编写好应用程序，然后放到云计算平台上运行。用户不需管理或控制底层的网络、服务器、操作系统、存储系统等，但要控制应用程序，可能还需要设置应用程序的运行环境。

从服务生产商或消费者的观点看，关于 PaaS 的观点至少有两种：

生产 PaaS 的某个人可能通过集成 OS、中间件、应用软件甚至是一个随后作为服务提供给客户的开发环境来生产一个平台。例如，开发 PaaS 的人可能会使其以一组 Sun xVM 管理程序虚拟机为基础，这组虚拟机包括一个 Net-Beans 集成化开发环境、一个 Sun GlassFish Web 栈，并支持像 Perl 或 Ruby 这样的其它编程语言。

使用 PaaS 的人会看到一个封装式服务，该服务是通过 API 提供给用户的。客户通过 API 与该平台互动，而且该平台执行一切必要的操作来管理和扩展其本身，以提供规定的服务水平。虚拟设备可以归类为 PaaS 的实例。例如，一个内容交换设备会将其所有组成软件对客户隐藏起来，而只向客户提供用来配置和部署服务的一个 API 或 GUI。

PaaS 的商业示例包括：

Google App Engine：谷歌推出了谷歌应用软件引擎(Google App Engine，简称 GAE)，这种服务让开发人员可以编译基于 Python 的应用程序，并可免费使用谷歌的基础设施来进行托管(最高存储空间达 500MB)。对于超过此上限的存储空间，谷歌按“每 CPU 内核每小时”10 ~ 12 美分及 1GB 空间 15 ~ 18 美分

的标准进行收费。最近，谷歌还公布了提供可由企业自定义的托管企业搜索服务计划。

Force. com 是 Salesforce. com 的企业云计算平台。Force. com 向企业提供

在云端上快速创建和实施业务应用程序所需的一切，包括数据库、无限的实时定制、强大的分析、实时工作流程和审批、可编程云逻辑、集成、实时移动部署、可编程的用户界面和网站功能。客户和合作伙伴在 Force. com 上创建了超过 120000 种定制应用程序来运营他们的业务，包括供应链管理、合规追踪、品牌管理、应收账款、索赔处理、休假应用程序以及其他更多的应用程序。

Microsoft Windows Azure Platform：Windows Azure Platform 是一套云端服务作业系统，作为 Windows Azure 平台的开发、服务代管及服务管理的环境。通过微软的 Windows Azure 系统可提供开发人员随选运算及存储，用来装载、延伸及管理网络上的 Web 应用程序、云端应用程序和网络服务，体系结构如图 1-4 所示。Windows Azure Platform 与 Visual Studio 进行了整合，支持一致性的开发经验。Windows Azure Platform 是个可同时支持微软及非微软程序语言及环境的开放性平台，起支撑技术如图 1-5 所示。微软欢迎 Eclipse、Ruby、PHP 与 Python 第三方工具和语言能够加入 Windows Azure Platform 行列。

NetSuite Suite 云：NetSuite 为其合作伙伴提供 Suite 云系统，在利润丰厚的



图 1-4 Windows Azure 平台架构

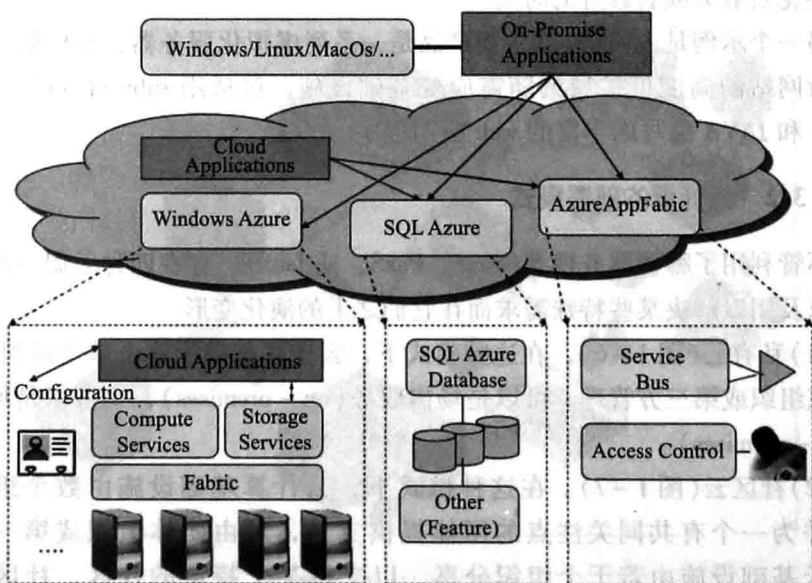


图 1-5 Windows Azure 支持的技术

云计算市场推出了自己的定制应用平台。据国外媒体报道，该平台提供的服务使其成为谷歌及甲骨文等公司的竞争对手，Suite 云平台将包含一系列的新产品、开发工具和服务，包括新的 Suite 云开发网络(SDN)，将专门面向独立软件开发商。独立软件开发商可以选择在 NetSuite 上或 Suite 云开发平台上利用