Broadview
www.broadview.com.cn

IBM

# 虚拟化与云计算

『虚拟化与云计算』小组 著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

IBM

# 虚拟化与云计算

主审：陈滢

著者：王庆波　金淬　何乐　赵阳　邹志乐　吴玉会　杨林

（以加入研究团队的时间先后为序）

電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

## 内 容 简 介

本书系统阐述了当今信息产业界最受关注的两项新技术——虚拟化与云计算。云计算的目标是将各种IT资源以服务的方式通过互联网交付给用户。计算资源、存储资源、软件开发、系统测试、系统维护和各种丰富的应用服务，都将像水和电一样方便地被使用，并可按量计费。虚拟化实现了IT资源的逻辑抽象和统一表示，在大规模数据中心管理和解决方案交付方面发挥着巨大的作用，是支撑云计算伟大构想的最重要的技术基石。本书以在数据中心采用服务器虚拟化技术构建云计算平台为主题，全面地勾画出虚拟化与云计算的产生背景、发展现状和关键技术等。本书体系完整，内容丰富，有助于广大读者理解信息产业今后发展的大脉络。



**图书在版编目（CIP）数据**

虚拟化与云计算 / 《虚拟化与云计算》小组著.—北京：电子工业出版社，2009.10
ISBN 978-7-121-09678-5

Ⅰ.虚… Ⅱ.虚… Ⅲ.①虚拟技术－研究②计算机网络－
研究 Ⅳ.TP391.9 TP393

中国版本图书馆CIP数据核字（2009）第183285号

策划编辑：郭立、刘皎
责任编辑：郭立
文字编辑：刘皎
印　　刷：北京机工印刷厂
装　　订：三河市鹏成印业有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　邮编100036
开　　本：700×1000　1/16　印张：17.25　字数：256千字
印　　次：2009年10月第1次印刷
印　　数：4000册　定价：45.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。
质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。
服务热线：（010）88258888。

# 目 录

## 第1章 数据中心的构建与管理



## 第2章 虚拟化概论

## 第3章 虚拟化的关键技术



## 第4章 虚拟化的业界动态

## 第5章 云计算概论



## 第6章 云架构

## 第7章 云计算的关键技术与挑战



## 第8章 云计算的业界动态

# 序一

在世界日趋变平变小的今天，每一个国家在享受全球化浪潮带来的机遇时，自我保护能力也随之降低。不然，源起美国不良信用房贷的一场经济危机何以给中国内地的中小企业造成影响？因世界的扁平，我们受益于在全球加速流动的各种资源，然而资源的分配始终是不平均的，且永远处于动态变化，我们时刻面临着挑战——创造更大的价值，拥有更多的资源。因此，我们必须学会好好地管理这个变化中的世界，或者说，世界需要变得更智慧，让我们继续享受全球化带来的好处，同时使环境资源得到更有效的利用、经济继续增长、实现人类的可持续性发展。这就是IBM公司在2009年初向全球提出的人类共同的愿景——“智慧的地球”。当高速发展的信息技术融入整个世界的运转，人类可以更透彻地感知这个世界，并实现全面的互联互通，所产生的海量信息转化为人类对世界更深刻的洞察，指导人类更智慧地管理地球上的一切系统，比如“智慧的能源”、“智慧的医疗”、“智慧的交通”、“智慧的城市”等。

信息技术自身同样需要变得更加智慧来应对复杂的未知世界。虚拟化与云计算作为“智慧的信息技术”的重要组成部分，已成为当今信息产业领域最受瞩目的新兴概念。虚拟化这项将引起信息技术变革、促使产业格局重新划分、改变企业和个人使用信息资源方式的先进技术越来越受到业界和科研部门的重视，云计算也从一个新兴事物逐渐渗透到信息产业的各个领域。在产业界，各大公司投入大量资源研究和开发云计算产品，其新兴技术和产品正在不断涌

现，传统的信息服务产品也在向云计算模式转型。在研究领域，学术会议纷纷增加了云计算专题。这一切都极大地推进了云计算技术的发展。

作为一家不断自我创新的百年企业，IBM公司以其对科技趋势、产业进步和世界发展的领先视角，一直在新技术的研究与应用领域走在世界的前沿。IBM中国研究院通过整合IBM全球的技术优势，采用先进的数学模型分析复杂的问题，再以虚拟化和云计算作为基础架构，实现实时、快速的计算和处理，从而形成对世界更深刻的认知和洞察，支持人类做出更准确的判断和预测，产生更有效的决策和反应。这种与实践结合的理念将研发部门和行业应用紧密联系起来，使研究成果对行业产生真正的价值。

IBM中国研究院在虚拟化与云计算领域从事了多年的研究工作，积累了丰富的经验。虚拟化与云计算这两项技术将对信息产业产生深远的影响，然而目前国内外还缺乏一本系统、深入地介绍相关技术的图书。我们不仅希望探索前沿的领域，创造更先进的技术，我们还希望将研究成果产业化，先进技术被广泛掌握，为中国社会的持续发展建立人才储备，本书正是基于这样的理念精心创作而成的。本书融入了我们在相关领域的经验，系统、清晰、全面地介绍了虚拟化和云计算的概念、架构、关键技术及最新研究动向，希望能帮助大家更好地了解虚拟化和云计算。

“智慧的地球”是一个美好的愿景，也赋予了我们光荣的使命——创造更多智慧的技术，培养更多智慧的人才，共建一个更加智慧的地球。

李实恭 博士
IBM大中华区首席技术官
IBM中国研究院院长

# 序二

“虚拟化”和“云计算”，这两个当下很时新，同时也的确是标志着计算机技术发展进入一个新阶段的概念，在本书中被具体地联系起来了。

我想，希望能搞清楚这两个概念及它们之间关系的人不会少，例如，我在2008年底教育信息存储暨校园下一代数据中心建设与应用大会上试着以“云计算、虚拟化、海量单增信息系统”为题讲过，2009年1月在北京大学深圳研究生院试着以“云计算、网格、虚拟化——概念及其关系”为题也讲过，但与现在这本书的角度和深度相比，那些都是皮毛了。

在本书中，虚拟化和云计算这两个抽象的概念，通过数据中心这个具体事物的构建与管理的需求有机地联系了起来。这是本书具有独特意义的要点。

读这本书，会发现简洁和实用是其鲜明的特点。篇幅不长，对要介绍的内容的层次把握得较好，几乎囊括了虚拟化和云计算所有重要的概念，但没有陷于过多的实现细节。在这个意义上，这本书是比较好读的，有一定计算机专业知识的人都可能饶有兴趣地读下去，得到的收获是对有关领域的宏观把握，这不仅对在业界把握技术走向的管理人员有用，对在大学把握研究方向的教授有用，而且对具体从事虚拟化和云计算技术研究与开发的人员也同样有用。

书中也有不少很有特色的具体内容。例如关于虚拟器件的阐述，我感到是本书的一个亮点。相关的内容零零碎碎在其他材料中也能看到一些，但明确地提出虚拟器件的管理是数据中心虚拟化的主要线索，并加以系统

的阐述，看来还是第一次。而通过几个实例来引入云计算，并尝试对其内涵与外延给出比较准确的刻画，与其他几个相关的概念进行区别，尽管不能说人人都会同意，也不一定都是很精辟，但所携带的信息和认识无疑对读者是很有帮助的。另外难能可贵的是，尽管作者都来自于IBM中国研究院，但书中对业界动态的介绍也体现了全面性和客观性。

本书的作者都是在第一线工作的青年研究人员，他们的工作背景一方面使得书中的内容体现了很强的实践性，同时字里行间也充满了对所从事工作的自豪和激情。在紧张的工作之余能花时间编写出这样一本很及时的书来，与广大读者分享他们的认识与经验，令人欣喜，我向他们表示祝贺。我相信这本书会使许多人受益，也祝愿我们的作者能在虚拟化和云计算技术的发展中不断有新的心得和贡献。

李晓明 博士
北京大学教授
北京大学网络所所长
2009年8月于北京大学

# 作者序

当我们写作者序时，本书的撰写已接近尾声，整个写作历程耐人回味。本书的作者大多是长期从事分布式计算和数据中心管理的研究人员，随着对虚拟化技术认识的逐渐加深，我们更加相信虚拟化技术将会在不远的将来给数据中心管理带来深刻的变革。怀着这样一份对未来的憧憬，我们于2005年在IBM中国研究院正式成立了虚拟化技术研究部。当时业界对虚拟化技术和大规模数据中心管理还缺乏深刻的认识，也未掌握成熟的方法，我们将研究重点放在应用虚拟化技术来简化服务部署、提高运行维护效率、降低管理复杂性、提升资源利用率，从而打造节能环保的数据中心。

经过几年的实践，我们开创了应用虚拟器件技术管理信息服务和数据中心的完整方法，其中部分成果已经成为IBM内部和产业界的标准；研发了一系列与之配套的管理工具，用于虚拟器件的制作、激活、部署、动态资源调度、运行时管理等。利用这些方法和工具，我们将凝聚了IBM多年经验的软件产品和最佳实践解决方案封装成基于虚拟器件的虚拟化解决方案，并通过快捷部署激活工具简化应用上线过程，为用户提供更稳定、更可靠的服务，为管理人员提供更简捷、更智慧的管理模式。

云计算是近年来兴起的新理念，目标是将计算和存储简化为像公共的水和电一样易用的资源，用户只要连上网络即可方便地使用，按量付费。云计算提供了灵活的计算能力和高效的海量数据分析方法，企业不需要构建自己专用的数据中心就可以在云平台上运行各种各样的业务系统，这种创新的计算模式和商业模式吸引了产业界和学术界的广泛关注。我们所从事的虚拟化研究是云计算的基石，是云计算最重要的支撑技术。凭借在虚拟化领域积累的经验，我们在2008年将研究范畴扩展到云计算，部门更名为虚拟化与云计算研究部，这给我们的研究工作提供了更大舞台，也提出了更多的挑战。作为IBM公司内部最早开展云计算研究的部门，在过去的几年里，我们在国内外的会议和杂志上发表论文十几篇，申请国际专利二十余项，研究成果已经融入到IBM的多款云计算产品和解决方案中。

作为长期工作在产业前沿的研究团队，我们到国内各大高校做了一些虚拟化和云计算的主题演讲，也发表了一些中文论文，并有部分英文论文被译成中文，但这些零散的资料很难系统地论述相关知识。确实，目前虚拟化技术仍处于普及阶段，需要人们更多地了解和接受，而云计算的概念就像它的名字本身一样，似乎仍被云雾笼罩，让人难识其真面目。于是，我们决定写一本专门介绍虚拟化和云计算的图书，让广大同行和读者了解本领域最新的技术成果，共同感受信息产业变革带来的机遇与挑战。在写作过程中，我们力求用严谨的语言来阐述概念，用科学的精神来介绍技术，从大局的角度介绍业界动态。在紧张的科研工作之余，我们齐心协力，终于完成了这本富有创新挑战的专业著作。

本书的写作由王庆波统筹协调和脉络把握，陈滢负责整体审阅和统稿，金淳负责项目管理。各章执笔者的分工如下：第1章金淳；第2章、第7章赵阳；第3章、第4章何乐；第5章邹志乐、金淳；第6章邹志乐；第8章吴玉会；杨林重新创作了第2章、第5章。

本书撰写历时半年多，其间经历了创作、审阅、讨论、修订、再审阅、再讨论、再修订等数次迭代。仅是打印装订成册的正式“审阅版本样书”就有7版之多，我们都为团队成员的敬业精神、创作激情、协作能力和执行力感到骄傲和自豪。拿到印刷前的最后清样之时，整个创作团队兴奋不已，这样一部凝聚了IBM中国研究院虚拟化与云计算研究部的心血之作，终于要和广大读者见面了。

作为全球第一本系统、全面介绍虚拟化与云计算的新著，它首次为广大读者勾勒出虚拟化和云计算的来龙去脉，揭示这些抽象、浪漫的名字背后的技术细节。如果本书能够为企业的技术主管和研发人员揭示未来信息产业的发展方向，能够将高校教师和学生带入一个新的科学技术领域，能够启发立志创业的人士找到时代赐予的机遇，我们将感到由衷的欣慰。

为了能把这些内容及时展现给读者，成书难免仓促，如有遗漏和纰误，请各位专家和读者不吝指教。希望广大读者能够从本书中获益。

# 前言

在过去的半个多世纪，信息技术的发展，尤其是计算机和互联网技术的进步极大地改变了人们的工作和生活方式。大量企业开始采用以数据中心为业务运营平台的信息服务模式。进入新世纪后，数据中心变得空前重要和复杂，这对管理工作提出了全新的挑战，一系列问题接踵而来。企业如何通过数据中心快速地创建服务并高效地管理业务？怎样根据需求动态调整资源以降低运营成本？如何更加灵活、高效、安全地使用和管理各种资源？如何共享已有的计算平台而不是重复创建自己的数据中心？业内人士普遍认为，信息产业本身需要更加彻底的技术变革和商业模式转型，虚拟化和云计算正是在这样的背景下应运而生的。

虚拟化技术很早就在计算机体系结构、操作系统、编译器和编程语言等领域得到了广泛应用。该技术实现了资源的逻辑抽象和统一表示，在服务器、网络及存储管理等方面都有着突出的优势，大大降低了管理复杂度，提高了资源利用率，提高了运营效率，从而有效地控制了成本。由于在大规模数据中心管理和基于互联网的解决方案交付运营方面有着巨大的价值，服务器虚拟化技术受到人们的高度重视，人们普遍相信虚拟化将成为未来数据中心的重要组成部分。

虽然虚拟化技术可以有效地简化数据中心管理，但是仍然不能消除企业为了使用IT系统而进行的数据中心构建、硬件采购、软件安装、系统维护等环节。早在大型机盛行的20世纪五六十年代，就是采用“租

借”的方式对外提供服务的。IBM公司当时的首席执行官Thomas Watson曾预言道：“全世界只需要五台计算机”，过去三十年的PC大繁荣似乎正在推翻这个论断，人们常常引用这个例子，来说明信息产业的不可预测性。然而，信息技术变革并不总是直线前进，而是螺旋式上升的，半导体、互联网和虚拟化技术的飞速发展使得业界不得不重新思考这一构想，这些支撑技术的成熟让我们有可能把全世界的数据中心进行适度的集中，从而实现规模化效应，人们只需远程租用这些共享资源而不需要购置和维护。

云计算是这种构想的代名词，它采用创新的计算模式使用户通过互联网随时获得近乎无限的计算能力和丰富多样的信息服务，它创新的商业模式使用户对计算和服务可以取用自由、按量付费。目前的云计算融合了以虚拟化、服务管理自动化和标准化为代表的大量革新技术。云计算借助虚拟化技术的伸缩性和灵活性，提高了资源利用率，简化了资源和服务的管理和维护；利用信息服务自动化技术，将资源封装为服务交付给用户，减少了数据中心的运营成本；利用标准化，方便了服务的开发和交付，缩短了客户服务的上线时间。

虚拟化和云计算技术正在快速地发展，业界各大厂商纷纷制定相应的战略，新的概念、观点和产品不断涌现。云计算的技术热点也呈现百花齐放的局面，比如以互联网为平台的虚拟化解决方案的运行平台，基于多租户技术的业务系统在线开发、运行时和运营平台，大规模云存储服务，大规模云通信服务等。云计算的出现为信息技术领域带来了新的挑战，也为信息技术产业带来了新的机遇。然而，真正系统、全面地阐述云计算概念和技术及虚拟化在云计算中的发展和应用的书却是寥寥无几。本书作为全球第一本介绍虚拟化和云计算的图书，正好弥补了这一空白，为对云计算和虚拟化技术感兴趣的人员讲述相关的知识和理论。

本书前4章着重介绍数据中心管理和虚拟化技术，后4章着重介绍云计算的概念和动态。下面简要介绍一下各章的主要内容。

第1章介绍了数据中心的构建与管理。首先讲述了数据中心的概念、历史和发展情况，随后介绍构建数据中心的最佳实践方法和数据中心的管理维护，最后分析了新一代数据中心的需求和挑战。

第2章对虚拟化技术进行了概述。首先介绍虚拟化技术的定义，以及常见的虚拟化技术；接着，鉴于服务器虚拟化的重要性，着重讨论服务器虚拟化的概念、支撑技术、特点、性能和优势；最后对其他类型的虚拟化技

术做了简要介绍。

第3章介绍虚拟化的关键技术。首先介绍如何创建虚拟器件和虚拟化解决方案；然后描述如何部署虚拟化服务，包括部署、激活虚拟器件及将现有服务迁移到虚拟化环境等；最后介绍了运行、维护虚拟化数据中心的关键技术。

第4章对虚拟化技术的业界动态进行了介绍，主要涉及IBM、VMware、Xen/Citrix和Microsoft等几个虚拟化厂商。内容涉及每个厂商的简介、产品线及产品的特性等。

第5章对云计算技术进行了概述。首先介绍云计算的概念，对云进行分类，而且为了使读者清晰地了解云计算，在后面还针对云计算与其他相似概念进行了辨析；然后分析云计算的优势及为信息产业带来的变革；最后讨论云计算产生的源动力。

第6章着重介绍云架构。定义云架构的不同层次，分析每个层次的核心功能和技术挑战，并通过示例加深读者对每个层次的理解。

第7章概述云计算的关键技术与挑战。介绍云计算中的关键技术，包括已有的研究成果和目前的发展状况，然后讨论了一系列经典问题在云计算中所面临的新挑战。

第8章介绍云计算的业界动态，主要涉及几个领先的云计算厂商，包括IBM、Amazon、Google、Salesforce.com和Microsoft。介绍每个厂商的云计算产品线，分析其产品的功能和特点，使读者能够对主要的云计算厂商和产品有个总体认识，对业界的最新动态有较为全面的了解。

在附录中我们列出了2009年超级计算机的世界排名。有兴趣的读者可以通过本书最后的参考文献获取更多的虚拟化和云计算的知识。

在编写本书时，我们力图使不同职业和背景的读者都能从本书中获益。

如果您是企业的技术负责人或数据中心运行维修人员，您将更深刻体会到虚拟化和云计算技术为企业IT部门、信息系统规划和数据中心运行维修带来的深刻变革。我们提供的技术讨论、产品比较和案例分析，将有助于您在脑海中勾画下一步的战略。

如果您是从业的技术研发人员，您能系统地了解虚拟化和云计算的产生背景、发展现状、技术要点和未来趋势。通过本书的梳理，能够更加准确地把握业界前沿的科技和理念，认清信息业界发展的大脉络，形成适用于产业未来的大局观。

如果您是大专院校计算机及相关专业的学生，您将获得无法从现有课本中得到的技术知识。本书将为您打开一扇通往未来的窗户，帮助您拓宽视野，完善知识结构，储备适用于未来信息产业的知识和技能。

本书适合于从头至尾阅读，也可以按照喜好和关注点挑选独立的章节阅读。我们希望本书的介绍能加深您对虚拟化与云计算的理解，获得您所期待的信息。

为了便于与广大读者进行交流，本书设立了博客和电子邮箱，分别是：

博客网址：http://blog.sina.com.cn/vccbook/

电子邮箱：vccbook@sina.com

我们将在博客中提供相应的幻灯片资料和可能存在的内容更新等。期待与您更进一步的交流。

# 致谢

本书从构思、写作、修订到出版，得到了许多同事和朋友的无私帮助，在此我们要对他们致以最衷心的感谢。

首先感谢IBM中国研究院优越和宽松的研究氛围，让我们在科研工作之余，能够抽出时间来撰写本书，这种以创新为本的公司理念激发出我们很多的创作灵感。同样，如果没有来自同事们的睿智见解和热情鼓励，我们根本就不可能完成此书。

感谢曲民慷慨地对本书的每一部分提出了宝贵的修订意见，并使本书的语言变得更加流畅；感谢张煜、尹大力、田瑞雄认真审阅了本书的大部分章节，并提出了深刻的见解。感谢多年来一直和我们工作在一起的吴朱华、晋普、温志广、陈靓等同事；感谢IBM中国研究院的蒋忠波、张剑鸣、张斌花费了宝贵的时间来阅读本书的部分章节，我们从他们的建议中获得了很多有益的帮助；感谢王浩所领导的团队使我们了解了更多的数据中心能耗管理方面的知识；感谢李影所领导的研究团队在远程管理和应用可伸缩性方面给予的热情支持；与沈晓卫在虚拟化与云计算方面的讨论使我们获益匪浅；感谢孙伟和郭常杰与我们分享了SaaS和多租户技术；正是与黄莹所领导的云平台研发团队的紧密合作，加深了我们在该领域的认识；感谢研究院系统维护部门的周英、许继涛等同事帮助我们了解数据中心的日常运营；感谢IBM全球技术服务部门的资深构架师杨杰使我们了解了更多的Web 2.0和云计算的案例，以及蔡妍和蒋聚旸与我们分享的网络架

构和数据中心构建方面的经验；感谢多年来一直和我们工作在一起的访问学生，他们的激情和活力为我们的工作增添了很多乐趣。

感谢多年来和我们工作在一起的IBM全球研发团队。通过和他们在虚拟化和云计算领域的合作，我们积累了大量的实践经验和对行业的认知，否则本书只能是纸上谈兵。他们是：软件部WebSphere品牌的WebSphere CloudBurst Appliance和WebSphere Application Server Hypervisor Edition研发团队，以及WebSphere性能分析团队；软件部战略组的OVF标准制定和虚拟器件设计部署团队；Watson研究中心的虚拟化运行时管理研究团队和IBM RC2研发团队；软件部Lotus品牌的虚拟化实施团队；软件部Tivoli品牌的TPM、ITM、TADDM、TSAM研发团队；软件部DB2品牌的DB2器件研发团队；系统与技术部的Director/Ensemble和Xen研发团队。

感谢一直以来身处政府、通信、教育、互联网等行业的合作伙伴，与他们的紧密合作使得我们一直走在时代的前沿。

感谢石贝贝帮助我们完成视频录制和部分文字整理工作。感谢汤竹、李国兰、张晓敏在出版流程上的大力协助。

我们要向电子工业出版社博文视点团队表示感谢。感谢郭立总经理的亲自审阅，并与我们分享对图书创作的理解，激发了我们出好书、出精品书的决心。感谢刘皎在创作、编辑、出版过程中对我们一如既往的热情支持。感谢与我们合作的编辑人员，他们细致耐心的工作使本书能够顺利出版。

还有很多人与我们分享了对虚拟化和云计算的理解，阅读了本书的审阅稿，并提出了宝贵的意见，在此特别感谢。

最后，感谢家人对我们一如既往的支持，他们是我们努力工作和快乐生活的动力源泉。

# 免责声明

读者在阅读本书内容之前，应仔细阅读本声明。凡以任何方式阅读或直接、间接使用本书内容者，均视为对本声明全部内容的认可和接受。

1. 本书所有内容仅代表本书作者的个人观点，与IBM公司的立场无关。IBM公司不对本书内容的准确性、可靠性或完整性提供任何明示或暗示的保证。对于任何因直接或间接采用、转载本书内容而造成的损失，本书作者和IBM公司均不承担责任。

2. 本书作者或IBM公司对本书所引用资料的版权归属和权利的瑕疵情况不承担核实责任。如任何单位或个人认为本书涉嫌侵犯其合法权益，应及时向本书作者提出书面意见并提供相关证明材料和理由，本书作者在收到上述文件后将采取相应措施。

3. 本书所引用的资料涉及到了非IBM公司产品，这些资料是从相应产品供应商所提供的说明或其他可公开获得的资料中获取的。本书作者或IBM公司没有对这些产品进行测试，也无法确认其性能的准确性、兼容性或任何其他关于非IBM公司产品的声明。有关非IBM公司产品的性能等问题应当向这些产品的供应商咨询。

4. 本书所引用资料的作者不因本书的引用行为而与本书作者或IBM公司之间产生任何关系或关联。

5. 本免责声明以及其修改权、更新权及最终解释权均属本书作者所有。

CHAPTER 1

# 第1章 数据中心的构建与管理

对于大多数人来说，“数据中心”是个略带神秘色彩的地方，没有窗户的高墙，恒定的温度和湿度，排列整齐的机架，跳跃闪烁的指示灯，海量的数据在这里穿梭，关键的业务在这里运行，这一切都充满科幻的色彩。但实际上，数据中心离我们每个普通人并没有那么遥远，甚至可以说是紧密相连。当你在银行办理业务时，整个交易的流程都在银行的数据中心中完成。当你在互联网上冲浪搜索信息时，请求都被搜索引擎的数据中心接收、处理和返回。如果说信息是血液，网络是血管，那么数据中心就是最关键的心脏，是信息世界的核心所在。

数据中心最早出现在20世纪60年代初。随着互联网的快速建设和信息技术的迅猛发展，到20世纪90年代中后期，数据中心进入了蓬勃发展期，建设规模和服务器数量每年都在以惊人的速度增长。本章将揭开数据中心的神秘面纱，向读者介绍数据中心的基本概念、核心功能、管理和维护工作，以及新一代数据中心的需求和挑战。

## 1.1 数据中心概述

数据中心是信息系统的中心，通过网络向企业或公众提供信息服务。

早期的数据中心主要指存放大型主机的机房。当时的大型主机非常昂贵，为了充分利用大型主机的资源，多个用户通过终端和网络连接到主机上来共享计算资源。20世纪80年代以后，计算机价格迅速下降，性能却反而提升，只要购买一台廉价的个人计算机，即可完成很多计算任务。到20世纪90年代，客户端/服务器的计算模式得到了广泛应用，用户安装客户端

软件后，通过互联网或局域网与服务器相互配合完成计算任务。在这种计算模式中，数据中心存放服务器（个人计算机所占的比重超过了大型机）并提供服务。互联网技术的蓬勃发展掀起了建设数据中心的高潮，不但政府机构和金融电信等大型企业扩建自己的数据中心，中小企业也纷纷构建数据中心，提供协同办公、客户关系管理等信息服务系统以支持业务的发展。信息系统为企业带来了业务流程的标准化和运营效率的提升，数据中心则为信息系统提供稳定、可靠的基础设施和运行环境，并保证可以方便地维护和管理信息系统。

最近几年，网上银行、网上证券和娱乐资讯等网络服务逐渐普及，网络用户数量的不断攀升也促进了各种规模数据中心的涌现，数据中心的发展进入了鼎盛时期。

数据中心在逻辑上包括硬件和软件。硬件是指数据中心的基础设施，包括支撑系统和计算机设备等；软件是指数据中心所安装的程序和提供的服务。图1.1展示了数据中心的逻辑示意图。一个完整的数据中心在其建筑之中，由支撑系统、计算机设备和信息服务这三个逻辑部分组成。支撑系统主要包括电力设备、环境调节设备和监控设备，这些系统是保证上层计算机设备正常、安全运转的必要条件。计算机设备主要包括服务器、存储设备和网络设备等，这些设施运行着上层的信息服务。信息服务的质量依赖于底层支撑系统和计算机设备的服务能力。只有整体统筹兼顾，才能保证数据中心的良好运行，为用户提供高质量、可信赖的服务。





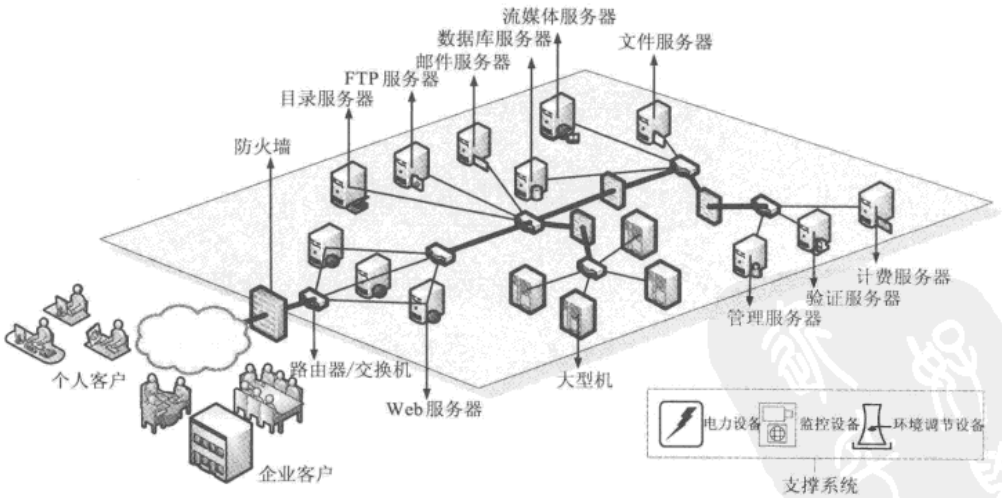



图1.1　数据中心逻辑示意图

## 1.2 数据中心的设计和构建

数据中心的设计和构建是一项系统工程，相关人员需要相互协作来完成总体设计、建筑和基础设施的构建，以及软硬件的采购和上线。本节将为读者介绍这些工作和相关的流程。

### 1.2.1 总体设计

数据中心的设计是一个系统、复杂、迭代的过程。数据中心设计者要在特定预算的情况下，让数据中心能够满足公司现有及将来不断增长的业务需求。数据中心的设计过程需要各类参与者不断地协商，平衡多方面的因素，比如在预算的限制和数据中心的性能间进行平衡。通常情况下，设计阶段决定了落成后数据中心的质量。合理的评估规划、全面周详的设计是构建数据中心关键的第一步。

从20世纪60年代初开始，世界各地的工程人员在构建数据中心的过程中不断总结，形成了系统的数据中心建设标准，如我国的国家标准《电子信息系统机房设计规范》（GB 50174-2008）和美国的《数据中心电信基础设施标准》（TIA-942）。这些标准为数据中心的设计，尤其是建筑、机电、通风等基础设施的规划提供了基本的依据。除了有标准可以依据，设计人员还可以参考以往工程中积累下来的实践经验，以现实需求为基础，合理运用新技术，提高数据中心的管理效率和整体性能。

构建数据中心需要遵守一些核心设计理念，即简单、灵活、可扩展、模块化。遵守这四个理念可以使得数据中心的设计清晰、高效、有条理。其中，简单的理念要求设计容易被理解和验证；灵活的理念保证数据中心能不断适应新的需求；可扩展的理念使数据中心能够随着业务的增长而扩大；模块化的理念是将复杂的工程分解为若干个小规模任务，使设计工作可控而易管理。

长期以来，业界采用等级划分的方式来规划和评估数据中心的可用性和整体性能。采用这种方法可以明确设计者的设计意图，帮助决策者理解投资效果。美国Uptime Institute提出的等级分类系统已经被广泛采用，成为设计人员在规划数据中心时的重要参考依据。在该系统中，数据中心按照

其可用性的不同，被分为四个等级（Tier）。

- 第一等级（Tier Ⅰ）被称为“基础级”（Basic Site Infrastructure），该级别的数据中心没有冗余设备（包括计算和存储），所有设备由一套线路系统（包括电力和网络）相连通。
- 第二等级（Tier Ⅱ）被称为“具冗余设备级”（Redundant Capacity Components Site Infrastructure），该级别数据中心具有冗余设备，但是所有设备仍由一套线路系统相连通。
- 第三等级（Tier Ⅲ）被称为“可并行维护级”（Concurrently Maintainable Site Infrastructure），该级别数据中心具有冗余设备，所有计算机设备都具备双电源并按照数据中心的建筑结构合理安装。此外，Tier III要求数据中心拥有多套线路系统，任何时刻只有一套线路被使用。
- 第四等级（Tier Ⅳ）被称为“容错级”（Fault Tolerant Site Infrastructure），该级别数据中心具有多重的、独立的、物理上相互分隔的冗余设备，所有计算机设备都具备双电源并按照数据中心的建筑结构合理安装。此外，Tier IV要求数据中心拥有动态分布的多套线路系统来同时连通计算机设备。

可见，随着等级的提高，数据中心具有了更强的可用性和整体性能。目前，已落成的数据中心在进行升级改造时都在力争达到Tier IV的要求。而面向云计算的下一代数据中心在设计时更是以Tier IV作为建设的标准。





## 1.2.2 建筑的设计与构建

构建一个数据中心有多种方式，究竟采用什么方式取决于企业的发展战略和预算。租用机房对于资金较少的公司是一个不错的选择，这样可以节省建设机房及管理维护数据中心的成本。对于需要拥有独立数据中心的企业，可以选择利用现有的建筑构建数据中心或者设计构建一个新的建筑作为数据中心。数据中心的建筑在安全、高度和承重方面都有严格的标准，无论是利用现有的建筑还是构建新的建筑都需要考虑数据中心构建标准。

构建数据中心，面临的第一个问题就是选址。选址要综合考虑多种因素，包括公司发展战略、预算、运营成本和安全等诸多因素，其中

通信、电力和地理位置是选址的三个主要考虑因素。光纤通信技术的发展解决了信息的长距离、高带宽快速传递的问题，因此，数据中心的选址不存在服务半径的问题，只要能够方便地接入主干通信网，即可服务于全球的客户。电力供应是构建数据中心需要考虑的另一个因素，数据中心所在位置必须能够提供充足、稳定的电力供应，并且电力成本足够低，因为电力是数据中心长期运营成本中的一大笔开销。为了提供可靠、稳定的服务，数据中心对可靠性和可用性都有严格的要求，所以选择地理位置的时候，安全是必须考虑的因素，应该尽量远离核电站、化工厂、飞机场、通信基站、军事目标和自然灾害频发的地带。

其次，构建数据中心需要考虑建筑要求，包括建筑的规模、布局、高度、地板的承重能力和室内布局等。数据中心的规模取决于企业的需求和预算，这个直接关系到能承载多少服务器，以及将来可以扩展到多大的规模。数据中心可以小到一个房间，大到一层楼甚至是整幢楼房。布局的设计要考虑到各个房间的大小、分布、面积和功能等，比如要考虑如何设置配线间、服务器存放区域和管理员房间等。良好的布局能够提高制冷效率，降低制冷成本。数据中心对楼层的高度有一定要求，除了机架占用的空间外，还要保留足够的高度，在地板下进行网线和电线的布线，在屋顶布置照明、防火、安全监控等设施。数据中心的服务器一般比较密集，楼板的承重能力通常要高于普通建筑，因此在设计楼板的承重能力时需要综合考虑数据中心的容量，包括服务器的数量、制冷设备等相关辅助设备的数量。此外，数据中心对室内环境要求较高，许多设备对温度、湿度和灰尘都有特定的要求，通常要避免室内设有窗户。

数据中心设计完成后，就进入了施工阶段，也就是根据设计实现数据中心的阶段。与建造其他建筑类似，施工阶段有许多烦琐的工作需要处理。为了保证工程质量，需要有专门的监管部门控制施工进度，并根据设定的标准进行阶段性验收，项目完成后还需要进行全面验收才能交付。

### 1.2.3 基础设施的设计与构建

为了确保设备的正常运行，网络、电力和环境控制设施等基础设施是必不可少的。如图1.2所示，电力是数据中心运行的动力，网络保证了服务器及存储的互联和访问，环境控制设施为设备运行提供了合适的温度、湿度等环境条件。基础设施的设计同IT设备的规模是紧密相关的。比如服务器的数量

直接影响所需要的电量；服务器数量越多，释放出来的热量会随之增长，制冷设备也需要相应增加。为了使IT设备相互连接，网络设施的设计建造同样是至关重要的。下面详细介绍上面三种基础设施的设计和构建。

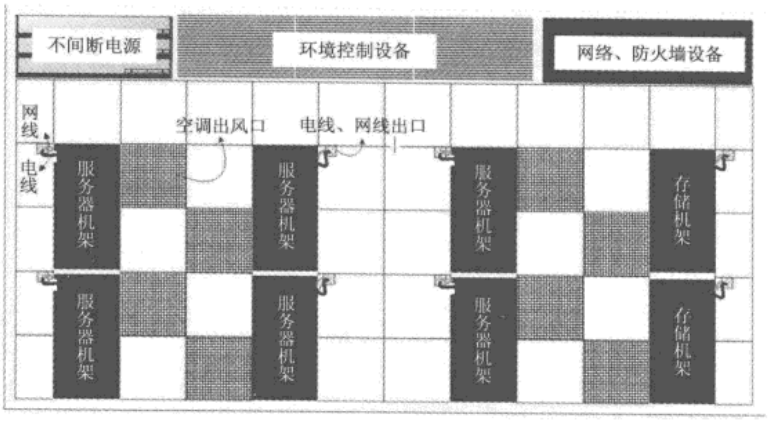



图1.2　数据中心基础设施示意图

电力系统的设计是数据中心基础设施设计中最为关键的部分，关系到数据中心能否持续、稳定的运行。电力系统的设计需要考虑数据中心的电力负荷限制、电力公司和冗余配备、电力设施的布局。数据中心内的电力负载主要有照明用电、消防应急系统用电、计算机设备用电和制冷设备用电。由于业务的重要性，以上各项电力负载均需要冗余来保证其可用性。在电力负荷确定后，数据中心的规划等级决定了电力冗余设备的配置。举例来说，Tier IV数据中心的电力系统可靠性需要达到99.99%，意味着平均每5年才会发生一次电力事故，平均每年电力事故引起的宕机时间为0.8小时。应对这样的可用性要求，数据中心需要采用市电双路供电，设置双总线UPS（Uninterruptible Power Supply，不间断电源）冗余，延时15分钟，同时配备柴油发电机作为第二重备份，在市电仍未恢复且UPS耗尽前及时接入。在数据中心的设计中，电力线路和插座的布局也是很重要的。数据中心内部IT系统和环境控制设备等基础设施（比如服务器、交换机及空调等）的分布直接影响电力线路的布局。设计线路布局还需要考虑将来扩展的需求及支持设备的类型，不同国家的设备对电压、电流的要求也是有差异的。此外，数据中心的电力系统还需要进行机房接地系统和防雷接地系统的设计，保证数据中心的电力安全。

网络基础设施的设计与电力系统的设计类似，需要与企业的业务需求紧密结合，主要包括网络供应商的选择和内部网络拓扑的设计。现在多数业务都支持通过互联网进行访问，所以业务的可用性和服务质量在一定程

度上取决于网络供应商的服务质量。如果业务对网络服务质量的要求比较高（比如银行ATM服务），则需要考虑多家网络供应商接入。一般数据中心的网络包含至少三级结构：网络供应商的网络接入连接到数据中心的核心交换机；二级交换机向上连接到核心交换机，向下同数据中心的机架互连；机架内部的服务器则通过机架内置的网络交换模块同二级交换机连接。每级交换机的性能和出口、入口的带宽选择都与数据中心内部的负载分布密切相关。

环境控制设施保证了数据中心的设备有一个适宜的运行环境，包括温度、湿度及灰尘的控制。设计环境控制设施需要考虑IT设施的规模、服务器的类型和数量等。温度控制作为环境控制中最为重要的问题已经被广泛研究，现在数据中心常采用的制冷方式有：风冷、水冷和机架内利用空气—水热交换制冷等。依据Tier IV标准，数据中心要求具有双路冷源和双冗余管路系统。如图1.3所示，为了布线的方便，一般都将机房地板架空，利用这个空间铺设网络线路、电力线路，以及将冷气分发到数据中心的每个角落。精密空调通过循环吸收热空气，制造冷气。在机架的前方，通过镂空的地板，将冷气送入机架，冷气流经机架带走服务器的热量，转换成热空气从机架后面重新流入制冷装置的进风口。风冷的设计有两个关键点：一是热通道和冷通道的设计，要避免热空气流入服务器机架中；二是单位时间送给每个机架的冷气必须能够满足整个机架的需求，否则机架下层的服务器排出的热空气就可能向上流动，使机架上层的服务器不能获得良好的制冷效果。风冷的一个主要问题是制冷能力有限，所以机架内服务器的密度不能太大。机架内空气—水交换制冷能有效提高机架内机器密度。随着绿色数据中心概念的推广，节能已经是数据中心设计的一个重要目标，水冷在节能和制冷效果方面都具有明显的优势，正被越来越多的数据中心采用。

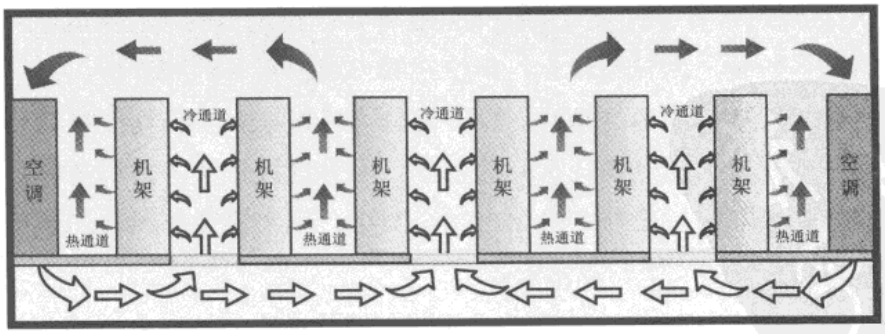



图1.3　数据中心风冷示意图

## 1.2.4 数据中心上线

数据中心上线包括以下几个步骤：选择服务器、选择软件、机器上架及软件部署和测试。下面将分别介绍这些步骤。

选择服务器需要综合考虑多方面因素，比如数据中心支持的服务器数量及数据中心将来要达到的规模和服务器的性能等。由于服务器是主要的耗电设备，所以节能也是一个重要的考虑因素。数据中心的服务器按照类型可以分为塔式服务器、机架式服务器和刀片服务器这三大类。下面将分别介绍这三种最常见的服务器。

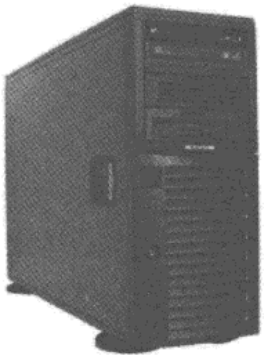


图1.4 塔式服务器

**塔式服务器**的外观与个人计算机的主机差不多，如图1.4所示。与普通PC相比，塔式服务器的主板可扩展性较强，接口和插槽比普通PC多一些，机箱的尺寸比普通PC稍大。塔式服务器成本较低，能够灵活地定制，可以满足入门级服务器的需求，所以应用范围非常广泛。不过塔式服务器也有其局限性：由于扩展性有限，塔式服务器很难满足规模较大的并行处理应用的要求；另外，由于占用空间较大，也不便于挪动和管理。

**机架式服务器**是一种外观按照统一标准设计的、配合机柜使用的服务器，如图1.5所示。由于采用统一的机架式结构，服务器可以方便地与其他网络设备连接，简化了机房的布线和管理。机架式服务器的尺寸有统一的标准：服务器的宽度为19英寸，高度以U为单位（U是表示服务器外部高度的单位，是Unit的简称，1U=1.75英寸，由美国电子工业协会确定）。通常标准的服务器高度在1U至7U之间，机柜的高度从22U至42U不等。图1.5描绘了从1U到4U四种不同尺寸的机架式服务器。相比于塔式服务器，机架式服务器的优点是占用空间较小，单位空间可放置更多的服务器，且管理方便。机架式服务器的不足是对制冷要求较高。机架式服务器广泛适用于服务器第三方托管（如电信托管）的企业，因为这种托管的费用常常是按照机器的空间收取的。另外，由于占用空间小，机架式服务器适用于服务器数量较大或者空间有限的数据中心。

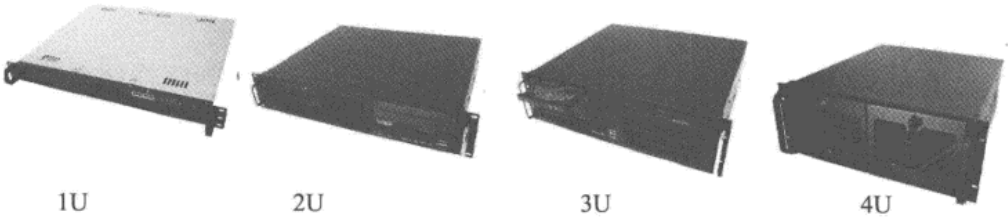



图1.5　机架式服务器

**刀片服务器**是在标准高度的机箱上插装多个卡式的服务器单元，由于这些服务器单元的外观很薄，故得名刀片服务器，如图1.6所示。实际上，每一块“刀片”都是一个独立的服务器，包括系统主板、硬盘、内存等设备，可以通过板载硬盘启动操作系统。若干刀片服务器连接起来，就形成了一个集群服务器，由所在的机箱提供高速的网络环境，同时共享机箱中的其他资源，协同完成计算任务。刀片服务器支持热插拔，这大大降低了系统维护的成本。刀片服务器比机架式服务器更加节省空间，光驱、显示器和制冷装置都是共享的，在一定程度上减少了成本。刀片服务器一般应用于大型数据中心或者计算密集的领域，如电信、金融行业和互联网数据中心等。对于企业和互联网服务提供商来说，随着业务的发展和对服务器需求的增长，刀片服务器在节约空间、便于管理、可扩展性方面拥有显著的优势，将成为未来服务器的主流产品。

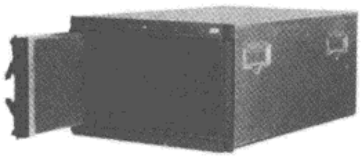


图1.6　刀片服务器

数据中心的软件主要包括操作系统、数据中心管理监控软件和与业务相关的软件（中间件、邮件管理系统、客户关系管理系统等软件）。

目前数据中心服务器操作系统主要有三大类：UNIX系统、Windows系统和Linux系统。数据中心要根据具体的业务需求选择适合的操作系统。

UNIX是一种技术成熟、可靠性高、安全性高的多任务分时操作系统。UNIX可满足政府机构和各行业大型企业的需求，适合运行企业的重要业务，是主流的企业IT操作平台。UNIX系统最早的雏形在1969年诞生于AT&T贝尔实验室，当时UNIX的所有者AT&T公司发布了UNIX的源码，许多机构在这个UNIX雏形的基础上进行了改进，产生了若干个UNIX的变种版本，如AIX、FreeBSD。UNIX系统常常与硬件配套，比如采购了IBM的

小型机就应选用AIX系统，从而达到最佳的系统性能。

Linux是一套可以免费使用和自由传播的、开源的类UNIX操作系统，由世界各地成千上万的程序员设计和实现。Linux系统在x86架构上实现了UNIX的主要特性，因而得到众多爱好者的广泛采用。Linux的发行版众多，常见的发行版有Ubuntu、SUSE和Redhat等。

Windows是Microsoft公司开发的操作系统，用于服务器的操作系统有Windows Server 2003和Windows Server 2008等。

数据中心大多以Web的形式向外提供服务，Web服务一般采用三层架构，从前端到后端依次为表现层、业务逻辑层和数据访问层。三层架构目前均有相关中间件的支持，如表现层的HTTP服务器，业务逻辑层的Web应用服务器，数据访问层的数据库服务器。主要产品有IBM公司的WebSphere（HTTP服务器，Web应用服务器）和DB2（数据库服务器），开源的产品有Apache（HTTP服务器）、Tomcat（Web应用服务器）和MySQL（数据库服务器）等。

数据中心的管理和监控软件种类繁多，功能涵盖系统部署、软件升级、系统、网络、中间件及应用的监控等。比如IBM的Tivoli系列产品和Cisco的网络管理产品等，用户可以根据自己的需要进行选择。

机器上架和系统初始化阶段主要完成服务器和系统的安装和配置工作。首先将机架按照数据中心设计的拓扑结构进行合理摆放，服务器组装完成后进行网络连接，最后安装和配置操作系统、相应的中间件和应用软件。这几个阶段都需要专业人员的参与，否则系统可能无法发挥最大的性能，甚至不能正常工作。举例来说，数据库软件安装完成后，需要根据服务器的硬件配置及应用的需求进行性能调优，这样才能最大程度地发挥数据库系统的性能。目前已经有了一些系统管理方案，支持自动地进行系统部署、安装和配置，这在一定程度上减少了技术人员的工作复杂度，简化了系统初始化的流程，提高了系统部署的效率。

服务器和软件安装配置完成后，就要开始对整个系统进行联合测试，检验软件是否正常运行、网络带宽是否足够，以及应用性能是否达到预期等。这个阶段需要参照设计阶段的文档逐条验证，测试系统是否满足设计要求。

## 1.3 数据中心的管理和维护

数据中心的管理和维护包含很多工作，涉及多种角色，包括系统管

理员、应用管理员、硬件管理员、机房管理员、数据管理员和网络管理员等，每个角色都不可或缺。在中小规模的数据中心里，经常一人身兼若干角色。本节将介绍数据中心管理和维护的主要工作。

### 1.3.1 硬件的管理和维护

硬件的管理和维护包括对硬件的升级、定期维护和更新等。业务规模的增长和系统负载的增加要求对服务器进行升级以适应业务发展的需要。系统运行一段时间后要定期对硬件进行检查和维护，保证硬件的稳定运行。当服务器发生硬件故障时，需要及时检测和定位故障，更换发生故障的部件。

升级或者更换部件时，不但要考虑服务器内各种部件的兼容性，还要协调这些部件的性能，消除性能瓶颈。服务器的CPU频率、内存大小、磁盘容量、I/O性能、网络带宽和电源供给能力等要达到均衡和协调，才能避免浪费并且使系统整体性能达到最优。在选取组件时，应尽量选取同一品牌和型号的组件，这样做一方面可以提高不同服务器组件之间的可替换性和兼容性，另一方面可以减少由于组件型号不同而对系统性能产生的影响。

灰尘是导致服务器故障的一个重要因素，服务器的散热风扇在运转时容易将尘土带入机箱，尘土中夹带的水分和腐蚀性物质附着在电子元件上，会影响散热或产生短路，增加系统的不稳定性。因此，定期的清理除尘是必不可少的。

### 1.3.2 软件的管理和维护

数据中心的常见软件包括操作系统、中间件、业务软件和相关的一些辅助软件，其管理和维护工作包括软件的安装、配置、升级和监控等。

操作系统的安装主要有两种方式：通过系统安装文件安装和克隆安装。安装文件的优势是支持多种安装环境和机器类型，但是安装中大多需要人工干预，容易出错，而且效率较低。对同一类服务器，则可以采用镜像克隆方式安装，避免手动安装引入的错误，减少人为原因引起的配置差异，提高部署效率。系统升级需要遵守严格的流程，包括新补丁的测试、验证及最后在整个数据中心进行规模分发和安装。补丁的分发有两种方式：一种是“推”方式，由中央服务器将软件包分发到目标机器上，然后

通过远程命令或者脚本安装；另一种是“拉”方式，在目标机器上安装一个代理，定期从服务器上获取更新。

安全性是操作系统管理和维护的重要内容，常见的措施包括安装补丁、设置防火墙、安装杀毒软件、设置账号密码保护和检测系统日志等。遵循稳定优先的原则，服务器一旦运行在稳定的状态，应避免不必要的升级，以免引入诸如软件和系统不兼容等问题。中间件和其他软件的管理和维护工作与操作系统类似，包括软件的安装、配置、维护和定期升级等。虚拟化技术的发展简化了软件的安装和配置工作，这部分内容将在后面的章节中进行详细介绍。

### 1.3.3 数据的管理和维护

数据是信息系统最重要的资产。事实上，构建信息系统的目标就是对数据的管理，保证数据安全、有效和可用。采用有效的数据备份和恢复策略能保证企业数据的安全，即使在灾难发生后，也能快速地恢复数据。数据中常常包含企业的商业机密，因此数据维护是数据中心维护工作的重中之重。随着信息技术的快速发展，数据量正在呈指数级增长。2003年全球人均数据量仅为0.8GB，2006年即上涨至24GB，预计到2010年将达到128GB，如此快速的增长趋势给数据维护带来了更大的挑战。

数据管理和维护主要包括数据备份与恢复、数据整合、数据存档和数据挖掘等，下面将逐一介绍这些内容。

数据备份是指创建数据的副本，在系统失效或数据丢失时通过副本恢复原有数据。数据备份的种类包括文件系统备份、应用系统备份、数据库备份和操作系统备份等。数据库备份应用最为广泛，主流的数据库产品都提供数据备份和恢复功能，支持不同策略的数据备份机制，并在需要时将系统数据恢复到备份时刻。目前数据库技术已经相当成熟，商业数据库软件的功能也很强大，管理员可在数据库中设置定时备份，也可以通过某种事件触发备份或者手动备份，使用起来很方便。例如，IBM DB2数据库支持完全备份和增量备份两种策略，实际使用中两者可以结合使用。为了保证数据安全，备份数据应存储在和原数据不同的物理介质上，以规避物理介质损坏所产生的风险。

数据整合通过将一种格式的数据转换成另一种格式，达到在多个系统之间共享数据和消除冗余的目的。一些企业由于历史原因拥有多个信息系

统，各个系统承担不同的功能，在某种程度上又和其他系统有交叉，数据整合可以满足这些系统间的数据共享需求。

数据归档是指将长期不用的数据提取出来保存到其他数据库的过程。数据挖掘是从归档数据库中分析寻找有价值的信息的过程。在业务系统运行过程中，会时刻产生新的业务数据，随着数据量的不断增大，数据库的规模越来越庞大，如果不能有效地处理这些数据，数据库的访问效率就会变差，进而影响业务系统的性能。归档的数据库也被称为数据仓库，可以为企业经营决策提供数据依据。保存在数据仓库中的数据一般只能被添加和查找，不能被修改和删除。归档时可按需对数据进行一些处理：首先清洗数据，去除错误或无效的数据；其次精简数据，将数据中可用于统计分析的信息抽取出来，将无用的信息删除，从而减少存档数据量，数据精简往往需要进行数据格式的转换。

### 1.3.4 资源管理

负载均衡是资源管理的重要内容，数据中心管理和维护时应做到负载均衡，以避免资源浪费或形成系统瓶颈。系统负载不均衡主要体现在以下几个方面。

第一，同一服务器内不同类型的资源使用不均衡，例如内存已经严重不足，但CPU利用率仅为10%。这种问题的出现多是由于在购买和升级服务器时没有很好地分析应用对资源的需求。对于计算密集型应用，应为服务器配置高主频CPU；对于I/O密集型应用，应配置高速大容量磁盘；对于网络密集型应用，应配置高速网络。

第二，同一应用不同服务器间的负载不均衡。Web应用往往采用表现层、应用层和数据层的三层架构，三层协同工作处理用户请求。同样的请求对这三层的压力往往是不同的，因此要根据业务请求的压力分配情况决定服务器的配置。如果应用层压力较大而其他两层压力较小，则要为应用层提供较高的配置；如果仍然不能满足需求，可以搭建应用层集群环境，使用多个服务器平衡负载。

第三，不同应用之间的资源分配不均衡。数据中心往往运行着多个应用，每个应用对资源的需求是不同的，应按照应用的具体要求来分配系统资源。

第四，时间不均衡。用户对业务的使用存在高峰期和低谷期，这种

不均衡具有一定的规律，例如对于在线游戏来说，晚上的负载大于白天，白天的负载大于深夜，周末和节假日的负载大于工作日。此外，从长期来看，随着企业的发展，业务系统的负载往往呈上升趋势。与前述其他情况相比，时间不均衡有其特殊性：时间不均衡不能通过静态配置的方式解决，只能通过动态调整资源来解决，这给系统的管理和维护工作提出了更高的要求。

总之，有效的资源管理方式能提高资源利用率，合理的资源分配能够有效地均衡负载、减少资源浪费、避免系统瓶颈的出现、保障业务系统的正常运行。

### 1.3.5 安全管理

作为企业信息系统的心脏，数据中心的安全问题尤为重要。数据中心的安全包括物理安全和系统安全。为了保证物理安全，数据中心需要配备完善的安保系统，该系统应实现7×24小时实时监控和录像、人员出入控制、人员远距离定位和联网报警功能。管理人员和授权用户可以随时随地接入系统获得相应的监控信息和回放资料。

系统安全主要是防止恶意用户攻击系统或窃取数据。系统攻击大致分为两类：一类以扰乱服务器正常工作为目的，如拒绝服务攻击等；另一类以入侵或破坏服务器为目的，如窃取服务器机密数据、修改服务器网页等，这一类攻击的影响更为严重。数据中心需要采取安全措施，有效地避免这两类攻击，常见的安全措施有以下几种。

第一，给服务器的账号设定安全的密码。账号和密码是保护服务器的最重要的一道防线，设定的密码要有足够的长度和强度，最好是数字、字母和符号的混合、大写和小写字母的混合，避免使用名字、生日等容易被猜中的密码，并且定期更换。

第二，采用安全防御系统，包括防火墙、入侵检测系统等。防火墙可以防止黑客的非法访问和流量攻击，将恶意的网络连接挡在防火墙之外。入侵检测系统可以监视服务器的出入口，通过与常见的黑客攻击模式匹配，识别并过滤入侵性质的访问。此外，网络管理员与安全防御系统配合可以进一步提高安全系数。管理员需要熟悉路由器、交换机和服务器等各种设备的网络配置，包括IP地址、网关、子网掩码、端口、代理服务器等，了解网络拓扑结构，在发现问题后迅速定位。网络管理员还要根据不

同IP和端口的访问流量统计，识别出非正常使用的情况并加以封禁。

第三，定时升级，及时给系统打补丁。不存在没有漏洞的系统，系统中的漏洞很多都隐藏在深处，不易被发现。一旦某个系统漏洞被黑客发现，就会对此类系统进行攻击或开发针对此类系统的病毒。与此同时，系统的开发者也会尽快发布补丁。攻击与防御，是一场速度的比拼。系统使用者要争取在第一时间安装系统补丁，不给黑客和病毒可乘之机。

第四，关闭不必要的系统服务。黑客可能通过有漏洞的服务攻击系统，即使无法通过这些服务攻击，开启的服务也可以给黑客提供信息，因此应该关闭不必要的服务。

最后，保留服务器的日志。虽然保留日志无法直接防止黑客入侵，但管理员可以根据日志分析出黑客利用了哪些系统漏洞、在系统安装了哪些木马程序，以便快速定位和解决问题。

## 1.4 新一代数据中心的需求

数据中心为信息服务提供运行平台，对新一代数据中心的需求从根本上源于对新一代信息服务的需求。随着信息服务在数量和种类上的快速增长，企业纷纷把核心业务和数据放到IT系统中运营。与此同时，用户数量也在不断攀升，用户对信息服务的依赖越来越强，企业和个人都需要更安全可靠、易于管理、成本低廉的信息服务。对信息服务的更高要求指明了新一代数据中心的发展方向，下面将从合理规划、流程化、可管理性、可伸缩性、可靠性、降低成本和节能环保这七个方面分别讨论。

### 1.4.1 合理规划

数据中心的建设是一项系统工程，从规划到设计，从选址到建设，从计算机设备到制冷系统，从网络安全到灾难防备，无一不需要合理规划。一个数据中心通常可以运行三十年左右，要使得数据中心在这三十年的时间内始终保持经济的运行状态，有很多复杂的因素需要考虑。比如需要考虑各种设备的更新换代，计算机设备通常以五年为更换周期，制冷系统的寿命可达十年以上，更新时需要合理选择设备，使用过度超前的设备或迟迟不更新都不能达到最经济的效果。再比如需要考虑设备冗余量，设备冗余可以提升系统的可用性，保证个别设备出现故障时整个系统仍能正常

运转。但是过多冗余会导致设备长期闲置、资源浪费，因此规划时需要具体分析，保证增加的冗余设备可以切实提高系统的可用性。

然而，由于企业难以预测IT系统的需求变化，有一些问题不能在设计数据中心时做出准确的规划。一方面，企业的整体运营越来越依赖于IT平台，而这些IT系统的负载并非长期不变，往往随着业务的发展而快速增长。有些企业甚至难以预见一年以后业务发展会带来怎样的系统负载变化。另一方面，IT系统的触角正逐渐伸展到企业业务和管理的各个角落，新上线的系统层出不穷，很难预测旧的管理方式和系统何时会被新系统取代。此外，IT系统本身越来越复杂，不可预见性也变得越来越强。这些变化的发生难以预测，一旦发生，数据中心的IT基础架构将无法支撑，急需扩容。同样，为难以预测的负载准备大量冗余也是不可取的。

综上所述，搭建数据中心需要合理规划各个环节，以保证数据中心在较为经济的状态下运营。同时，业务的动态性和不确定性会给数据中心的准确规划带来挑战。





### 1.4.2 流程化

通过合理规划和系统构建，落成后的数据中心需要为信息服务提供高效、可靠、稳定的运行环境和平台。因为信息服务的质量和成本是客户最关注的问题，信息服务管理自然成为数据中心的一项基本工作，其重要性不言而喻。信息服务管理的含义是以信息服务的形式为客户创造价值的一套组织能力，这种能力以流程的形式贯穿信息服务的整个生命周期。信息服务管理的核心是通过信息流程的标准化，帮助企业根据业务目标实现创新的、可视的、自动的、可控的信息服务，提高企业的运行效率和服务质量，为用户创造最大价值。

20世纪80年代，英国政府认为行政机构的信息服务质量有待提高，于是任命英国中央计算机与电信局（Central Computer and Telecommunications Agency, CCTA）制定一套指导行政机构使用信息资源的方法。CCTA将英国各行业在信息管理方面的最佳实践总结归纳起来，制定了信息技术基础构架库（Information Technology Infrastructure Library, ITIL）。这套信息服务管理流程库在英国各行业中得到了广泛认可和应用，并逐步延伸到全球。

ITIL从出现起至今经历了三个版本：最初版本ITIL V1总结了一系列关

于信息资源使用的实践，形成了一套标准化、可计量的信息资源使用指导规范；ITIL V2在ITIL V1的基础上进行了重新组织和完善扩充，形成了一套清晰的信息实践指导流程；ITIL V3是目前最新的版本，是对ITIL V1和ITIL V2的重构和丰富，融入了新的时代元素，突出了服务的核心地位。ITIL V3由三大部分组成：核心组件、补充组件和网络组件。核心组件涵盖了服务从创建到下线每个阶段的任务、目标及流程，由它们构成了通用的最佳实践。补充组件对不同行业领域的具体状况进行了深入探讨和剖析，并给出了专业的指导。网络组件是对前两个组件的扩充，提供了一个供用户学习、交流和发布信息的在线平台。

ITIL V3以服务为核心，覆盖了服务管理的整个生命周期，包括服务战略、服务设计、服务转换、服务运营和服务改进五个阶段，形成了富有生命活力的信息服务管理实践框架。下面将分别介绍这五个阶段的主要任务和目标。

服务战略的任务是了解现状、认清目标和设定规划。首先需要的是获取公司的资产、业务发展计划、职能部门和流程、市场和人员等信息，通过分析这些信息得到可以满足客户需求、为客户创造价值的服务目标，然后对贯穿整个服务生命周期的策略、指南和流程进行整体规划。

服务设计是对服务战略的实现，该实现依据的是服务战略中对服务设计和开发的描述，以及相关服务管理的流程定义，包括服务组合管理、服务级别管理、服务连续性和可用性管理等方面。

服务转换指的是采用有效的、低风险的方式将服务投入到运行环境中，还包括了对服务的变更、配置、测试、发布和评价等管理，同时将对整个过程中积累下来的知识进行组织和管理。

服务运营将最终实现服务战略的目标，服务运营需要保证服务交付的效果和支持和效率，从而实现客户和服务供应商的价值。这个阶段要保证服务的稳定性和可靠性，能满足服务设计变更及业务不断发展的需求。

持续的服务改进则是推动服务生命周期运转的源动力，通过在服务战略、设计、转换和运营方面进行改革创新，为客户提供更高质量的服务，在保证服务质量的前提下降低运营商的运营成本，从而达到客户和运营商双方的利益最大化。服务改进还涉及怎样将服务战略、服务设计、服务转换及服务运营同服务改进的效果有效关联，从而形成一个良性循环系统。

ITIL V3生命周期的核心框架是以服务战略为指导，以服务改进为原

动力，来推进设计、转换、运营三个阶段的迭代和螺旋上升，从而促进信息服务管理的改进，满足业务不断发展的需求。服务和业务在这种框架中结合得更为紧密，充分体现了以创造客户价值和降低运营成本为目标的理念，形成了一个不断发展、优化的信息服务管理生态系统。

ITIL作为信息服务管理标准化的最佳实践，有效保证了信息服务质量。由于在信息服务管理方面的优势，它被广泛应用于世界各地的数据中心。实施ITIL有助于规范企业的流程，明确不同部门的角色和职责，增进业务部门与IT部门的沟通，提高信息服务的可靠性、可用性和灵活性，降低信息服务管理的风险，从而降低企业的管理成本。对用户而言，ITIL贯彻了以用户为中心的理念，规范了明晰的服务标准和业务流程，不仅有利于保证服务质量，而且方便了用户使用信息服务，提高了用户的满意度。

### 1.4.3 可管理性

可管理性（Manageability）是指一个系统能够满足管理需求的能力及管理该系统的便利程度。系统管理是一个非常广泛的概念，包括全面深入地了解系统的运行状况、定期做系统维护以降低系统故障率、发现故障或系统瓶颈并及时修复、根据业务需求调整系统运行方式、根据业务负载增减资源，以及保证系统关键数据的安全等。大多数系统管理任务由系统管理员通过使用一系列管理工具来完成，少数管理任务需要领域专家的参与，另外一些任务可由管理系统自动完成。令人遗憾的是，很多数据中心由于没有管理工具而导致管理功能的缺失，还有一些管理系统或工具存在设计缺陷，导致系统的管理复杂烦琐。具体来说，数据中心的可管理性需求包含以下几个方面。

- 完备性保障了数据中心可以提供完整的管理功能集。数据中心包含种类繁多的软件和硬件设备，每个设备都要有相应的工具提供全面的管理支持，例如网络流量监控、数据库软件的参数配置、服务器所处环境温度监测等。
- 远程管理是指在远程控制台上通过网络对设备进行管理，免去了到设备现场进行管理的烦恼。
- 集成控制台将多个设备的管理功能集成起来，管理员可以在控制台上定义集成化的任务，通过一个指令完成对若干设备的协调控制，这简化了管理员的操作。

- 快速响应保障了发出的管理指令能够被尽快执行，即便执行指令需要较长的时间，也能较准确地把当前状态告知管理员，例如数据备份时需要显示备份的进度。
- 可追踪性保障了管理操作历史和重要的事务都能记录在案，以备查找。这些记录可以作为故障诊断的依据，帮助管理员或领域专家及时定位和解决问题。
- 方便性保障了管理功能对于管理员来说是真正可操作的，不会烦琐到无法承受的地步。这一方面要求将重复性的机械化的管理任务用工具替代而非手动完成，另一方面需要提供统一、简洁、直观的界面，管理员可以容易地找到被管理对象并发出管理指令。
- 自动化给可管理性提出了更高的要求，自动化程度越高，管理员的负担越小。自动化一般采用事件驱动模式，即当特定事件发生时采取特定的行动，若无法通过程序处理，则应立即发出警报通知管理员。很多管理系统都实现了一定程度的管理自动化，例如自动化故障诊断、定期自动检查磁盘空间、超过临界值时发出警报消息等。

### 1.4.4 可伸缩性

可伸缩性（Scalability）是指一个系统适应负载变化的能力，在负载变大的时候提高自身的能力以适应负载。例如，一个银行的营业厅可以在等候办理业务人数较多的时候开启更多的服务窗口，而人少的时候仅开启一两个窗口。一个可伸缩的算法可以容易地适应大规模的问题，一个可伸缩的计算机系统可以容易地通过增加硬件来提高吞吐量。

数据中心需要具备高可伸缩性的IT基础架构，可伸缩性可以从“伸”和“缩”两个角度理解。“伸”在信息服务上线运行或需要更多资源的时候及时、适量地给予资源分配，保证业务的正常运行不受影响。“缩”在信息服务下线或资源需求减少的时候适时回收资源，保证系统的资源高效利用，从而节省运营成本。

高可伸缩性的需求主要源于以下几点。首先，用户对服务的使用呈现规律性的高峰期和低谷期，虽然这种规律一定程度上可以预测，但仍然存在较大波动。其次，突发事件会对信息服务的负载造成难以预测的影响，

例如一个网络上流行的新闻、图片或视频，可以使相关网站的负载达到平时的百倍甚至千倍以上。此外，信息服务的使用量会随着业务的发展而增长，长期来看呈现上升的趋势。最后，新的服务层出不穷，对资源的需求也难以预测。

新一代数据中心对高可伸缩性的要求是及时、适量、细粒度、自动化和预动性。及时讲求的是快速反应，一旦发出指令后能在较短时间完成伸缩；适量需要分配给信息服务合适的资源；细粒度要求能以CPU、内存、磁盘为单位分配资源，而不是以物理服务器为单位，细粒度是适量分配的基础；自动化是指可以在一个控制台上，通过简单的操作完成为信息服务增加资源或服务器等工作，不需要人工进行准备机器、连接电缆、安装软件等烦琐的操作；预动性是指能有效预测出信息服务负载的变化趋势，并在负载增加之前就做好准备，以防负载变化后资源不足，对业务运行造成影响。





### 1.4.5 可靠性

可靠性（Reliability）是指一个组件或系统执行其功能的能力，系统成功完成指定功能的概率是衡量系统可靠性的常用指标。系统的可靠性取决于组成系统的组件本身的可靠性及组件之间的连接关系。组件之间常见的连接方式有串联、并联、K/N表决系统和混合连接，这几种连接方式构成了可靠性分析的基本模型。如果系统以串联方式连接，任意一个组件失效则整个系统失效；如果系统以并联方式连接，全部组件失效时整个系统才失效；K/N表决系统包含N个组件，当且仅当不少于K个组件失效时整个系统失效；复杂系统一般以上述几种方式组合的形式连接。

可靠性对数据中心的重要性不言而喻，在设计数据中心时应尽早考虑。从理论上讲，数据中心各层组件之间呈串联关系，联合起来为信息服务提供支撑，一旦某一层的组件失效，就可能导致信息服务的失效。提高可靠性的主要方法有故障避免和故障容错。故障避免是指提高单个组件的可靠性，减小其失效的概率。要做到故障避免需要研究组件失效的机理，如寿命失效、设计失效等，并针对不同的失效机理分别应对。故障容错是指增加冗余组件，利用组件之间的并联关系提升系统的可靠性，例如增加备份电源等手段。

目前数据中心可靠性分析出现了一些新的趋势，对可靠性的认识也更加深刻。首先，将可靠性与可维护性结合起来考虑，对于可维修或容易维修的故障，分析其修复率、平均修复时间等指标。一般来讲，对容易修复的故障容忍程度要高于难修复或不可修复的故障。其次，要重视对故障系统的管理，因为发生故障时信息服务停止运行的总时间为等待维修时间与维修时间之和，等待维修时间则取决于故障管理水平，如果管理水平低下，停机时间将会大大超过维修时间。再次，考虑故障的可容忍性时要对故障引发的后果的严重程度进行综合分析，以区分致命故障、严重故障和轻度故障。最后，需要用多种指标从不同维度来衡量可靠性，例如目前普遍认为使用无维修连续工作时间比单纯用失效概率来衡量可靠性对数据中心的管理人员更有实际意义。

### 1.4.6 降低成本

企业在IT系统上的投入逐年增多，20世纪70年代，普通的美国公司大约用10%的资本预算来购买信息技术，而30年以后这一比例已经上升到45%。许多企业因此不堪重负，他们普遍希望IT部门减少开支和提升效率，降低成本已经成为当前面临的大问题。此外，IT系统数量和规模的快速增长也使数据中心成本问题显得更为突出。

数据中心的成本构成分为一次性成本和运营成本。一次性成本主要包括建筑成本、服务器采购成本和其他设备采购成本；运营成本主要包括电力消耗和管理维护成本。服务器采购、电力消耗和管理维护成本是最主要的三项开支。20世纪60年代，计算机是非常昂贵的设备，一台大型主机的月租金可达几万美金，相比之下，其他成本都显得微不足道。随着IT产业的发展，尤其是x86处理器广泛普及以后，计算机在几十年之间变成了廉价的设备。随着处理器频率的不断提高，单处理器的能耗不断增加，经历了时代的变迁，电力消耗和管理维护的成本占数据中心成本的比例越来越高。图1.7为数据中心的成本构成及发展趋势。一方面，在过去几年中，企业的服务器数量在快速增加。另一方面，虽然企业用于采购服务器的开销基本维持不变，但是数据中心装机规模的增长使得管理和维护工作的复杂度迅速增加，管理成本和能耗也随之增大。

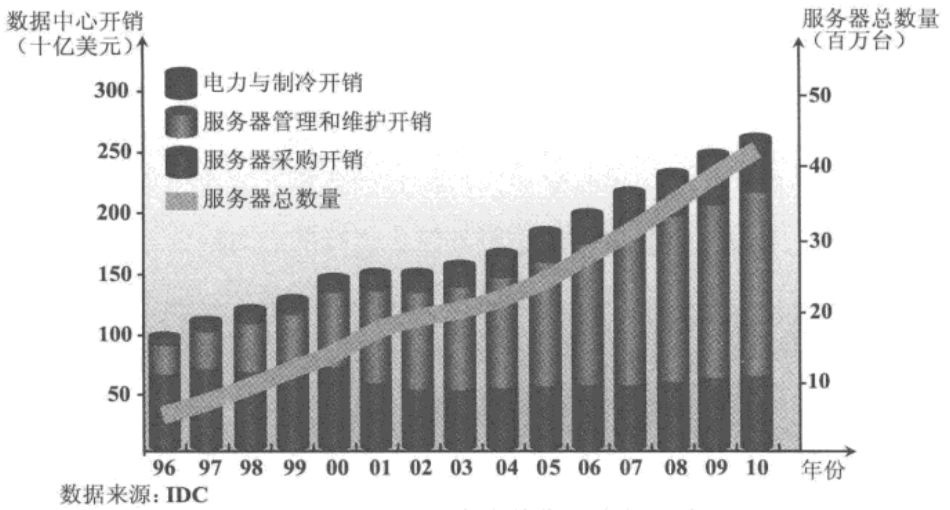



图1.7　数据中心成本构成及发展趋势

降低服务器的采购成本需要合理规划服务器更新换代的周期。IT设备降价较快，一旦服务器闲置，就会造成无形折旧，增加数据中心成本。因此规划时要结合业务需求，尽可能保证服务器的高利用率。

平均每个管理员可管理的服务器数量是评价数据中心管理维护是否高效的重要标准。当数据中心规模较小时，少数管理员即可承担管理维护任务，对管理维护水平的要求也相对较低。随着数据中心规模的增大，这种人力密集型的管理手段难以应付，使用专业的数据中心管理软件、工具和科学的方法可以大幅提升管理效率。

### 1.4.7 节能环保

美国环境保护署在2007年8月提交的一份报告中指出，全美数据中心的能源消耗在2006年占美国能耗的1.5%，预计到2011年将会增加一倍，节能环保已经成为IT基础设施建设中日益重要的话题。从经济角度来看，国际能源商品价格长期以来处于不断上涨的趋势中，随着企业对IT基础设施建设的投入不断加大，IT系统的能耗也随之攀升，摆在企业CIO们面前的一大问题是如何打造绿色数据中心，通过节能减少开支。从环境角度来看，环保是每一个企业的社会责任，企业需要通过减少耗电量来减少碳排放量，减缓全球变暖的步伐。已经有一些政府对达到绿色节能环保标准的数据中心给予政策性补贴。然而令人遗憾的是，很多企业数据中心的耗电量、耗电结构仍然是一笔糊涂账，有的企业甚至将数据中心的电费账单和办公楼的电费账单混在一起，完全没有节能环保方面的考虑。

图1.8是一个典型的数据中心电力消耗的示意图。据美国能源部统计，电力输送到数据中心后，平均只有45%被IT设备使用，其他55%则用于冷却系统等耗电设备。用于IT设备的部分，只有30%被处理器所用，剩下的70%则用于电源、风扇、内存、磁盘等部件。处理器的平均负载只有5%～20%，剩余部分都被浪费了。

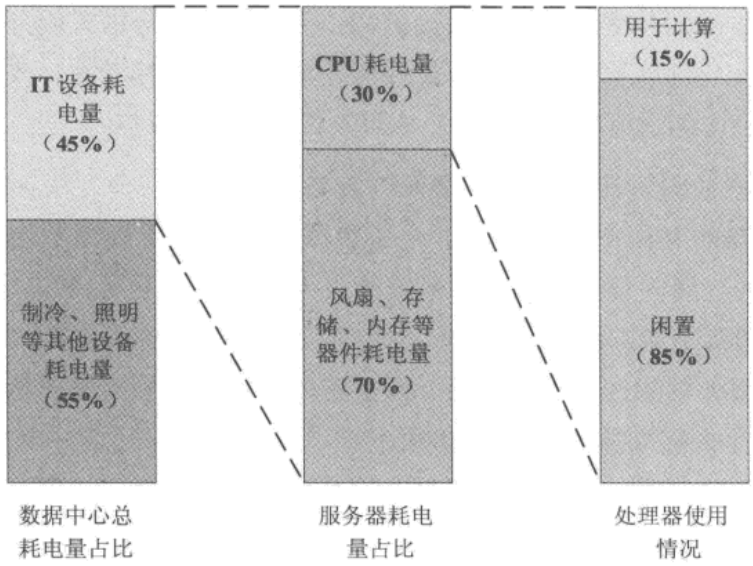



图1.8　数据中心耗电比例示意图

电能利用率（Power Usage Effectiveness，PUE）是在分析数据中心电力消耗时用到的重要概念，该标准由绿色网格联盟提出，已经成为国际上比较通行的衡量数据中心电力使用效率的指标。PUE的定义是总能耗与IT设备能耗的比值，是一个大于1.0的数值，PUE值越接近于1.0说明其他设备的能耗越小，效率也就越高。IT设备的能耗取决于IT设备的性能和业务负载，在不更新IT设备的前提下，其能耗由业务负载直接决定，由于短期内业务负载不会发生巨大改变，所以可以认为短期内IT设备的能耗是一个固定值。于是，总能耗与PUE值成正比，而PUE值取决于数据中心基础设施的设计和建设水平及所处环境的气候条件。据统计，国外先进的数据中心PUE值可达1.6～1.8，而国内的数据中心PUE值平均在2.0～2.5之间，中小规模的数据中心PUE值更高，有的甚至在3.0以上。近几年新设计建造的数据中心，PUE值可以达到1.8左右。一个典型的PUE为2.2的数据中心，IT设备耗电量占45%，空调设备耗电量占45%，而照明等其他设备耗电量之和不过10%，可以看出，一半以上的电量都被空调等设备消耗了。

节省数据中心的电力消耗需要从两方面着手：一方面需要在保证业务系

统需求的前提下，尽量降低IT设备的能耗；另一方面需要降低PUE值，提高电力使用效率，因为数据中心的能耗等于IT设备能耗和PUE值的乘积。

降低IT设备的能耗需要定期更新设备。人们普遍存在这样一个误区，认为增加服务器的使用年限可以降低数据中心的成本，于是仍然使用一些早该淘汰的服务器。其实恰恰相反，落后的服务器能耗更高，占地面积更大，出现故障的几率也随之增大。例如，刀片服务器与塔式服务器和机架式服务器相比性能更高，仅从单位性能耗电量一项指标考虑，改用刀片服务器节省的电能就可以抵过购买刀片服务器所增加的成本。

提高服务器资源利用率是降低IT设备能耗的另一个方法。目前数据中心服务器的利用率普遍很低，企业数据中心服务器资源平均利用率在10%～30%之间，很多Windows系统的服务器利用率不足10%。无论如何这样的数据都让CIO们难以接受，他们不愿相信多一半以上的IT投资都在被浪费。服务器的性能越来越强，而被有效利用部分的比例却越来越小。要了解为什么提高服务器资源利用率可以省电，先来了解一下服务器利用率和能耗的关系。服务器的能耗通常可以分为两部分：一部分是CPU的能耗，这部分能耗和CPU的利用率直接相关，CPU的利用率越高则能耗越高；另一部分是主板、内存、网络等其他部件的能耗，这一部分能耗基本为固定值，只与服务器是否开机有关。举一个简单的例子，假设有三台服务器的CPU利用率都是10%，如果可以把上面的应用迁移到一台服务器上，关掉剩下的两台就可以省下这两台服务器的固定能耗，而运行的服务器的CPU能耗也不会增加太多。电力使用效率是消耗单位电能可提供的计算力，大致是CPU频率和服务器功率的比值。虚拟化技术使得多个虚拟机可以共享同一台物理机，从而达到提升服务器资源利用率的目的。虚拟化技术正在被越来越多的企业广泛采纳，已有很多成功案例。

降低PUE值需要对数据中心的制冷系统做合理的设计和优化。常见的降低PUE值的方法包括数据中心选址、合理设定服务器间隔和空调温度、集中冷却、水冷降温等。首先是数据中心选址，由于空调的能耗与室外温度密切相关，因此将数据中心建在温度较低的地区可以有效减少制冷系统的能耗。其次，需要合理设定服务器间隔和空调温度。服务器太密集不利于通风散热，服务器太稀疏会增大数据中心面积，从而影响制冷效果。设定空调温度的原则是够用即可，并非越低越好。再次，集中冷却方法是给机柜加上一个隔热门，将机柜内外的空气隔开，让空调的出风口直接将冷

风送到机柜内部，这样做的好处是不需要对整个机房全部进行冷却。最后，水冷降温是比用空调降温更节能环保的方法，可以作为制冷系统的补充，例如Google公司在美国俄勒冈州Dalles的数据中心就建在一个河边，利用河水对数据中心进行冷却，冷水温度升高后被送到室外自然冷却，这一循环过程几乎不消耗电能。

## 1.5 小结

数据中心是企业的信息中心，它通过网络向企业和公众提供信息服务。随着计算机产业和互联网的发展，数据中心作为信息服务的运行环境，在人们的生活中扮演着越来越重要的角色。

设计和构建数据中心是一项复杂而专业的系统工程，涉及总体规划、建筑、电力、网络、制冷、服务器、管理软件及应用软件等各个方面，需要遵循一定的规范和标准，借鉴以往的成功经验，各类人员相互协作方可顺利完成。

数据中心上线后，管理和维护工作同样重要，尽管很多管理工具可以辅助完成这些复杂而专业的工作，但是随着数据中心规模的扩大及对服务质量要求的提高，数据中心的复杂性和管理开销逐年增加，这都给新一代数据中心提出了更高的要求，如果合理规划，可以提高其可管理性、可伸缩性、可靠性，并在降低成本的同时实现节能环保。

本书的后续章节将介绍虚拟化和云计算，这两项技术将会在很大程度上解决新一代数据中心所面临的问题。

CHAPTER 2

# 第2章 虚拟化概论

虚拟化技术（Virtualization）是伴随着计算机技术的产生而出现的，在计算机技术的发展历程中一直扮演着重要的角色。从20世纪50年代虚拟化概念的提出，到20世纪60年代IBM公司在大型机上实现了虚拟化的商用，从操作系统的虚拟内存到Java语言虚拟机，再到目前基于x86体系结构的服务器虚拟化技术的蓬勃发展，都为虚拟化这一看似抽象的概念添加了极其丰富的内涵。近年来随着服务器虚拟化技术的普及，出现了全新的数据中心部署和管理方式，为数据中心管理员带来了高效和便捷的管理体验。该技术还可以提高数据中心的资源利用率，减少能源消耗。这一切，使得虚拟化技术成为整个信息产业中最受瞩目的焦点。

本章将讲解虚拟化技术的定义，重点介绍当前最重要的服务器虚拟化技术，对它的概念、支撑技术、优势特点及性能进行分析和阐述，并讨论在数据中心中被广泛采纳的其他虚拟化技术。

## 2.1 虚拟化的定义

### 2.1.1 走近虚拟化

虚拟相对于真实，虚拟化就是将原本运行在真实环境上的计算机系统或组件运行在虚拟出来的环境中。一般来说，计算机系统分为若干层次，从下至上包括底层硬件资源、操作系统、操作系统提供的应用程序编程接口，以及运行在操作系统之上的应用程序。虚拟化技术可以在这些不同层次之间构建虚拟化层，向上提供与真实层次相同或类似的功

能，使得上层系统可以运行在该中间层之上。这个中间层可以解除其上下两层间原本存在的耦合关系，使上层的运行不依赖于下层的具体实现。由于引入了中间层，虚拟化不可避免地会带来一定的性能影响，但是随着虚拟化技术的发展，这样的开销在不断地减少。根据所处具体层次的不同，“虚拟化”这个概念也具有不同的内涵，为“虚拟化”加上不同的定语，就形成不同的虚拟化技术。目前，应用比较广泛的虚拟化技术有基础设施虚拟化、系统虚拟化和软件虚拟化等类型。虚拟化是一个非常宽泛的概念，随着IT产业的发展，这个概念所涵盖的范围也在随之扩大。

比如，操作系统中的虚拟内存技术是计算机业内认知度最广的虚拟化技术，现有的主流操作系统都提供了虚拟内存功能。虚拟内存技术是指在磁盘存储空间中划分一部分作为内存的中转空间，负责存储内存中存放不下且暂时不用的数据，当程序用到这些数据时，再将它们从磁盘换入到内存。有了虚拟内存技术，程序员就拥有了更多的空间来存放自己的程序指令和数据，从而可以更加专注于程序逻辑的编写。虚拟内存技术屏蔽了程序所需内存空间的存储位置和访问方式等实现细节，使程序看到的是一个统一的地址空间。可以说，虚拟内存技术向上提供透明的服务时，不论是程序开发人员还是普通用户都感觉不到它的存在。这也体现了虚拟化的核心理念，以一种透明的方式提供抽象了的底层资源。





## 2.1.2 虚拟化的定义

“虚拟化”是一个广泛而变化的概念，因此想要给出一个清晰而准确的“虚拟化”定义并不是一件容易的事情。目前业界对“虚拟化”已经产生如下多种定义。

“虚拟化是表示计算机资源的抽象方法，通过虚拟化可以用与访问抽象前资源一致的方法访问抽象后的资源。这种资源的抽象方法并不受实现、地理位置或底层资源的物理配置的限制。”——Wikipedia，维基百科

“虚拟化是为某些事物创造的虚拟（相对于真实）版本，比如操作系统、计算机系统、存储设备和网络资源等。”——WhatIs.com，信息技术术语库

“虚拟化是为一组类似资源提供一个通用的抽象接口集，从而隐

藏属性和操作之间的差异，并允许通过一种通用的方式来查看并维护资源。”——Open Grid Services Architecture

尽管以上几种定义表述方式不尽相同，但仔细分析一下，不难发现它们都阐述了三层含义：

- 虚拟化的对象是各种各样的资源；
- 经过虚拟化后的逻辑资源对用户隐藏了不必要的细节；
- 用户可以在虚拟环境中实现其在真实环境中的部分或者全部功能。

本书将援引IBM对虚拟化的定义，并基于该定义对虚拟化进行讨论。

**虚拟化是资源的逻辑表示，它不受物理限制的约束。**

在这个定义中，资源涵盖的范围很广，如图2.1所示。资源可以是各种硬件资源，如CPU、内存、存储、网络；也可以是各种软件环境，如操作系统、文件系统、应用程序等。按照这个定义，我们能更好地理解上一小节提到的操作系统中的内存虚拟化。内存是真实资源，而硬盘则是这种资源的替代品。经过虚拟化后，这两者具有了相同的逻辑表示。虚拟化层向上隐藏了如何在硬盘上进行内存交换、文件读写，如何在内存与硬盘间实现统一寻址和换入换出等细节。对于使用虚拟内存的应用程序来说，它们仍然可以用一致的分配、访问和释放的指令对虚拟内存进行操作，就如同在访问真实存在的物理内存一样。

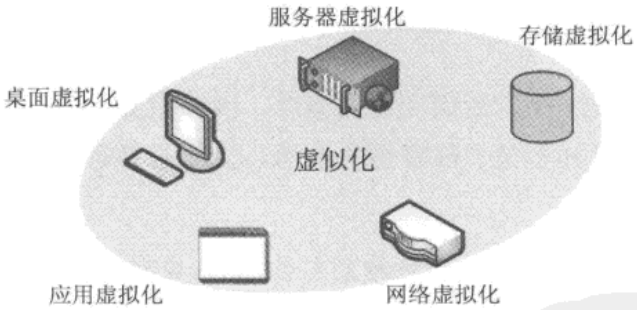



图2.1　包罗万象的虚拟化

虚拟化的主要目标是对包括基础设施、系统和软件等IT资源的表示、访问和管理进行简化，并为这些资源提供标准的接口来接收输入和提供输出。虚拟化的使用者可以是最终用户、应用程序或者是服务。通过标准接口，虚拟化可以在IT基础设施发生变化时将对使用者的影响降到最低。最终用户可

以重用原有的接口，因为他们与虚拟资源进行交互的方式并没有发生变化，即使底层资源的实现方式已经发生了改变，他们也不会受到影响。

虚拟化技术降低了资源使用者与资源具体实现之间的耦合程度，让使用者不再依赖于资源的某种特定实现。利用这种松耦合关系，系统管理员在对IT资源进行维护与升级时，可以降低对使用者的影响。

### 2.1.3 虚拟化的常见类型

在虚拟化技术中，被虚拟的实体是各种各样的IT资源。按照这些资源的类型分类，我们可以梳理出不同类型的虚拟化。目前，大家接触最多的就是系统虚拟化。比如使用VMware Workstation在个人电脑上虚拟出一个逻辑系统，用户可以在这个虚拟的系统上安装和使用另一个操作系统及其上的应用程序，就如同在使用一台独立的电脑。我们将该虚拟系统称做“虚拟机”，而VMware Workstation这样的软件就是“虚拟化软件套件”，它们负责虚拟机的创建、运行和管理。虽然虚拟机或者说系统虚拟化是当前最常使用的虚拟化技术，但它并不是虚拟化的全部。下面为读者介绍虚拟化的几种常见类型。

#### 1. 基础设施虚拟化

由于网络、存储和文件系统同为支撑数据中心运行的重要基础设施，因此本书将网络虚拟化、存储虚拟化归类为基础设施虚拟化。

网络虚拟化是指将网络的硬件和软件资源整合，向用户提供虚拟网络连接的虚拟化技术。网络虚拟化可以分为局域网络虚拟化和广域网络虚拟化两种形式。在局域网络虚拟化中，多个本地网络被组合成为一个逻辑网络，或者一个本地网络被分割为多个逻辑网络，并用这样的方法来提高大型企业自用网络或者数据中心内部网络的使用效率。该技术的典型代表是虚拟局域网（Virtual LAN, VLAN）。对于广域网络虚拟化，目前最普遍的应用是虚拟专用网（Virtual Private Network, VPN）。虚拟专用网抽象化了网络连接，使得远程用户可以随时随地访问公司的内部网络，并且感觉不到物理连接和虚拟连接的差异性。同时，VPN保证这种外部网络连接的安全性与私密性。

存储虚拟化是指为物理的存储设备提供一个抽象的逻辑视图，用户可以通过这个视图中的统一逻辑接口来访问被整合的存储资源。存储虚拟

化主要有基于存储设备的存储虚拟化和基于网络的存储虚拟化两种主要形式。磁盘阵列技术（Redundant Array of Inexpensive Disks, RAID）是基于存储设备的存储虚拟化的典型代表，该技术通过将多块物理磁盘组合成为磁盘阵列，用廉价的磁盘设备实现了一个统一的、高性能的容错存储空间。网络附加存储（Network Attached Storage, NAS）和存储区域网（Storage Area Network, SAN）则是基于网络的存储虚拟化技术的典型代表。

存储虚拟化是指把物理上分散存储的众多文件整合为一个统一的逻辑视图，方便用户访问，提高文件管理的效率。存储设备和系统通过网络连接起来，用户在访问数据时并不知道真实的物理位置。它还使管理员能够在一个控制台上管理分散在不同位置的异构设备上的数据。

### 2. 系统虚拟化

正如上文所述，目前对于大多数熟悉或从事IT工作的人来说，“虚拟化”这个词在脑海里的第一印象就是在同一台物理机上运行多个独立的操作系统，即所谓的系统虚拟化。系统虚拟化是被最广泛接受和认识的一种虚拟化技术。系统虚拟化实现了操作系统与物理计算机的分离，使得在一台物理计算机上可以同时安装和运行一个或多个虚拟的操作系统。在操作系统内部的应用程序看来，与使用直接安装在物理计算机上的操作系统没有显著差异。

系统虚拟化的核心思想是使用虚拟化软件在一台物理机上虚拟出一台或多台虚拟机（Virtual Machine, VM）。虚拟机是指使用系统虚拟化技术，运行在一个隔离环境中、具有完整硬件功能的逻辑计算机系统，包括客户操作系统和其中的应用程序。在系统虚拟化中，多个操作系统可以互不影响地在同一台物理机上同时运行，复用物理机资源。在下文，尤其是第4章中，读者将会接触到各种各样不同的系统虚拟化技术，比如应用于IBM z系列大型机的系统虚拟化、应用于基于Power架构的IBM p系列服务器的系统虚拟化和应用于x86架构的个人计算机的系统虚拟化。对于这些不同类型的系统虚拟化，虚拟机运行环境的设计和实现不尽相同。但是，在系统虚拟化中虚拟运行环境都需要为在其上运行的虚拟机提供一套虚拟的硬件环境，包括虚拟的处理器、内存、设备与I/O及网络接口等，如图2.2所示。同时，虚拟运行环境也为这些操作系统提供了诸多特性，如硬件共享、统一管理、系统隔离等。

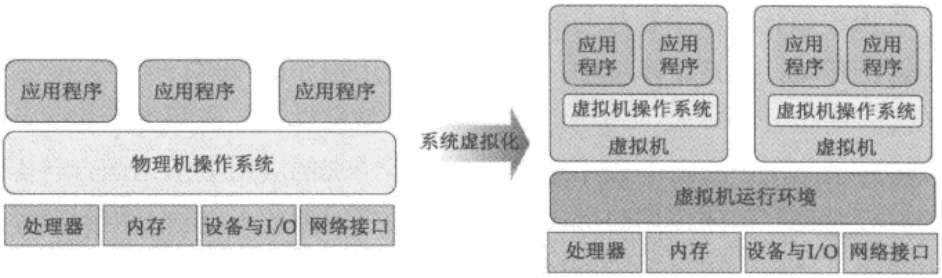



图2.2　系统虚拟化

相信很多读者都曾经或者正在将系统虚拟化技术运用到我们日常所用的个人电脑上。在个人电脑上使用系统虚拟化具有丰富的应用场景，其中最普遍的一个就是运行与本机操作系统不兼容的应用程序。例如，一个用户使用的是Windows系统的个人电脑，但是需要使用一个只能在Linux下运行的应用程序，他可以在个人电脑上虚拟出一个虚拟机并在上面安装Linux操作系统，这样就可以使用他所需要的应用程序了。

系统虚拟化更大的价值在于服务器虚拟化。目前，数据中心大量使用x86服务器，一个大型的数据中心中往往托管了数以万计的x86服务器。出于安全、可靠和性能的考虑，这些服务器基本只运行着一个应用服务，导致了服务器利用率低下。由于服务器通常具有很强的硬件能力，如果在同一台物理服务器上虚拟出多个虚拟服务器，每个虚拟服务器运行不同的服务，这样便可提高服务器的利用率，减少机器数量，降低运营成本，节省物理存储空间及电能，从而达到既经济又环保的目的。

除了在个人电脑和服务器上采用虚拟机进行系统虚拟化以外，桌面虚拟化同样可以达到在同一个终端环境运行多个不同系统的目的。桌面虚拟化解除了个人电脑的桌面环境（包括应用程序和文件等）与物理机之间的耦合关系。经过虚拟化后的桌面环境被保存在远程的服务器上，而不是在个人电脑的本地硬盘上。这意味着当用户在其桌面环境上工作时，所有的程序与数据都运行和最终被保存在这个远程的服务器上，用户可以使用任何具有足够显示能力的兼容设备来访问和使用自己的桌面环境，如个人电脑、智能手机等。

### 3. 软件虚拟化

除了针对基础设施和系统的虚拟化技术，还有另一种针对软件的虚拟化环境，如用户所使用的应用程序和编程语言，都存在着相对应的虚拟化概念。目前，业界公认的这类虚拟化技术主要包括应用虚拟化和高级语言虚拟化。

应用虚拟化将应用程序与操作系统解耦合，为应用程序提供了一个虚拟的运行环境。在这个环境中，不仅包括应用程序的可执行文件，还包括它所需要的运行时环境。当用户需要使用某款软件时，应用虚拟化服务器可以实时地将用户所需的程序组件推送到客户端的应用虚拟化运行环境。当用户完成操作关闭应用程序后，他所做的更改和数据将被上传到服务器集中管理。这样，用户将不再局限于单一的客户端，可以在不同的终端上使用自己的应用。

高级语言虚拟化解决的是可执行程序在不同体系结构计算机间迁移的问题。在高级语言虚拟化中，由高级语言编写的程序被编译为标准的中间指令。这些中间指令在解释执行或动态翻译环境中被执行，因而可以运行在不同的体系结构之上。例如，被广泛应用的Java虚拟机技术，它解除下层的系统平台（包括硬件与操作系统）与上层的可执行代码之间的耦合，来实现代码的跨平台执行。用户编写的Java源程序通过JDK提供的编译器被编译成为平台中立的字节码，作为Java虚拟机的输入。Java虚拟机将字节码转换为在特定平台上可执行的二进制机器代码，从而达到了“一次编译，处处执行”的效果。

本书主要讨论的是在数据中心设计实施虚拟化和构建云计算环境。围绕数据中心这个核心场景，本书对虚拟化技术的介绍有所侧重。首先，作为数据中心最主要的虚拟化技术，服务器虚拟化是我们讨论的重点。在2.2节，我们将着重介绍服务器虚拟化的概念、支撑技术、特点及优势。此外，一个完整的数据中心离不开网络和存储等基础设施。同样的，在交付应用时，软件虚拟化也会为数据中心的管理提供极大的便利。因此，网络和存储虚拟化，桌面与应用虚拟化作为数据中心的有机组成部分，将会在2.3小节介绍。





## 2.2 服务器虚拟化

### 2.2.1 基本概念

服务器虚拟化将系统虚拟化技术应用于服务器上，将一个服务器虚拟成若干个服务器使用。如图2.3所示，在采用服务器虚拟化之前，三种不同的应用分别运行在三个独立的物理服务器上；在采用服务器虚拟化之后，这三种应用运行在三个独立的虚拟服务器上，而这三个虚拟服务器可以被一个物理服务器托管。简单来说，服务器虚拟化使得在单一物理服务器上

可以运行多个虚拟服务器。服务器虚拟化为虚拟服务器提供了能够支持其运行的硬件资源抽象，包括虚拟BIOS、虚拟处理器、虚拟内存、虚拟设备与I/O，并为虚拟机提供了良好的隔离性和安全性。

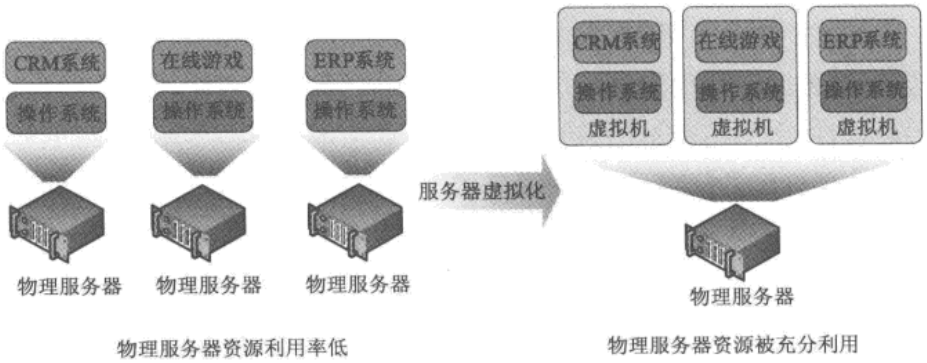



图2.3　服务器虚拟化

服务器虚拟化技术最早在IBM公司制造的大型机中使用，在20世纪90年代由VMware公司将其引入x86平台，并在2000年后迅速被业界接受，成为炙手可热的技术。由于看到服务器虚拟化应用在数据中心带来的巨大优势，各大IT厂商纷纷加大了对服务器虚拟化相关技术的投资：微软最新的服务器操作系统Windows Server 2008选装组件包含了服务器虚拟化软件Hyper-V，并承诺Windows Server 2008支持现有的其他主流虚拟化平台；2007年底，Cisco公司宣布通过购买股份的方式对VMware公司进行战略投资。多个主流Linux操作系统发行版，比如Novell公司的SUSE Enterprise Linux、RedHat公司的RedHat Enterprise Linux中都加入了Xen或KVM虚拟化软件，并鼓励用户安装使用。虚拟化技术被多家主流技术公司，包括Cisco、Google、IBM、Microsoft等列为技术和商业战略规划中的重点方向。

与此同时，众多企业的IT部门也在陆续实施虚拟化，将服务器虚拟化技术应用于企业数据中心。Forrest Research在2007年的一份报告中指出，约有40%的企业开始使用服务器虚拟化，服务器虚拟化厂商VMware也在2008年的用户大会上宣布，《财富》杂志列出的100强公司已经全部采纳了服务器虚拟化技术，《财富》杂志列出的500强公司中的绝大多数也使用了服务器虚拟化软件。

下面是几种使用最广泛的服务器虚拟化产品。

- Citrix公司的Xen。
- IBM公司的PowerVM、zVM。

- Microsoft公司的Virtual PC、 Virtual Server和Hyper-V。
- VMware公司的VMware Server、VMware Workstation、 VMware Player和VMware ESX Server。

在这些产品中，IBM公司的PowerVM和zVM是对应该公司的p系列服务器和z系列服务器的产品。这些服务器不同于x86体系结构，它们具有强大的硬件性能，并在设计之初就考虑到了如何虚拟出多台服务器以便充分利用服务器性能的问题。p系列服务器虚拟化技术PowerVM和z系列服务器虚拟化技术zVM就是为解决这一问题而产生的。这些技术从诞生至今发展了几十年，非常成熟和稳定。关于p系列服务器和z系列服务器的虚拟化技术将在第4章中详细讨论。

与p系列服务器和z系列服务器不同，x86架构在设计之初并没有考虑要支持服务器虚拟化技术，这使得在其之上实现服务器虚拟化相当困难。但是，随着x86服务器的广泛应用，以及其硬件能力的不断提高，实现x86系统上的服务器虚拟化的需求逐渐迫切。下面我们将着重介绍如何实现服务器虚拟化及其核心技术与典型方案。



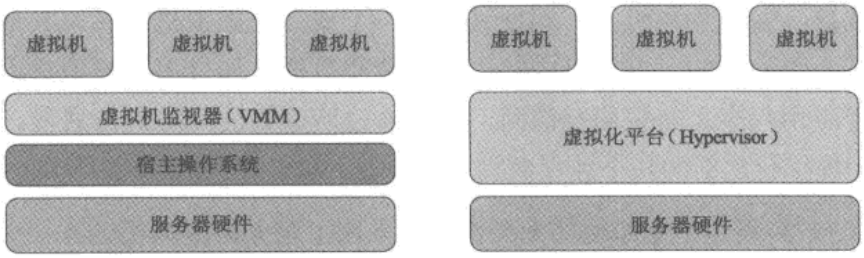


### 2.2.2 典型实现

服务器虚拟化通过虚拟化软件向上提供对硬件设备的抽象和对虚拟服务器的管理。目前，业界在描述这样的软件时通常使用两个专用术语，它们分别如下。

- 虚拟机监视器（Virtual Machine Monitor, VMM）。虚拟机监视器负责对虚拟机提供硬件资源抽象，为客户操作系统提供运行环境。
- 虚拟化平台（Hypervisor）。虚拟化平台负责虚拟机的托管和管理。它直接运行在硬件之上，因此其实现直接受底层体系结构的约束。

这两个术语通常不做严格区分，其出现源于虚拟化软件的不同实现模式。在服务器虚拟化中，虚拟化软件需要实现对硬件的抽象，资源的分配、调度和管理，虚拟机与宿主操作系统及多个虚拟机间的隔离等功能。这种软件提供的虚拟化层处于硬件平台之上、客户操作系统之下。根据虚拟化层实现方式的不同，服务器虚拟化主要有两种类型，如图2.4所示。表2.1比较了这两种实现方式。




（1）寄宿虚拟化　　（2）原生虚拟化

图2.4　服务器虚拟化的实现方式

**表2.1 服务器虚拟化的实现方式比较**

| | 寄宿虚拟化 | 原生虚拟化 |
|---|---|---|
| 是否依赖于宿主操作系统 | 完全 | 不 |
| 性能 | 低 | 高 |
| 实现的难易程度 | 易 | 难 |

- 寄宿虚拟化。虚拟机监视器是运行在宿主操作系统之上的应用程序，利用宿主操作系统的功能来实现硬件资源的抽象和虚拟机的管理。这种模式的虚拟化实现起来较容易，但由于虚拟机对资源的操作需要通过宿主操作系统来完成，因此其性能通常较低。这种模式的典型实现有VMware Workstation和Microsoft Virtual PC。
- 原生虚拟化。在原生虚拟化中，直接运行在硬件之上的不是宿主操作系统，而是虚拟化平台。虚拟机运行在虚拟化平台上，虚拟化平台提供指令集和设备接口，以提供对虚拟机的支持。这种实现方式通常具有较好的性能，但是实现起来更为复杂，典型的实现有Citrix Xen、VMware ESX Server和Microsoft Hyper-V。

### 2.2.3 关键特性

前面我们介绍了服务器虚拟化的概念及典型实现方式。无论采用以上何种方式，服务器虚拟化都需要具有以下特性，来保证可以被有效地运用在实际环境中。

（1）多实例。通过服务器虚拟化，在一个物理服务器上可以运行多个虚拟服务器，即可以支持多个客户操作系统。服务器虚拟化将服务器的逻辑整合到虚拟机中，而物理系统的资源，如处理器、内存、硬盘和网络等，是以可控方式分配给虚拟机的。

（2）隔离性。在多实例的服务器虚拟化中，一个虚拟机与其他虚拟机

完全隔离。通过隔离机制，即便其中的一个或几个虚拟机崩溃，其他虚拟机也不会受到影响，虚拟机之间也不会泄露数据。如果多个虚拟机内的进程或者应用程序之间想相互访问，只能通过所配置的网络进行通信，就如同采用虚拟化之前的几个独立的物理服务器一样。

（3）封装性。也即硬件无关性，在采用了服务器虚拟化后，一个完整的虚拟机环境对外表现为一个单一的实体（例如一个虚拟机文件、一个逻辑分区），这样的实体非常便于在不同的硬件间备份、移动和复制等。同时，服务器虚拟化将物理机的硬件封装为标准化的虚拟硬件设备，提供给虚拟机内的操作系统和应用程序，保证了虚拟机的兼容性。

（4）高性能。与直接在物理机上运行的系统相比，虚拟机与硬件之间多了一个虚拟化抽象层。虚拟化抽象层通过虚拟机监视器或者虚拟化平台来实现，并会产生一定的开销。这些开销即为服务器虚拟化的性能损耗。服务器虚拟化的高性能是指虚拟机监视器的开销要被控制在可承受的范围之内。



## 2.2.4 核心技术

服务器虚拟化必备的是对三种硬件资源的虚拟化：CPU、内存、设备与I/O。此外，为了实现更好的动态资源整合，当前的服务器虚拟化大多支持虚拟机的实时迁移。本节将介绍x86体系结构上这些服务器虚拟化的核心技术，包括CPU虚拟化、内存虚拟化、设备与I/O虚拟化和虚拟机实时迁移。

### 1. CPU虚拟化



CPU虚拟化技术把物理CPU抽象成虚拟CPU，任意时刻一个物理CPU只能运行一个虚拟CPU的指令。每个客户操作系统可以使用一个或多个虚拟CPU。在这些客户操作系统之间，虚拟CPU的运行相互隔离，互不影响。

基于x86架构的操作系统被设计成直接运行在物理机器上，这些操作系统在设计之初都假设其完整地拥有底层物理机硬件，尤其是CPU。在x86体系结构中，处理器有4个运行级别，分别为Ring 0、Ring 1、Ring 2和Ring 3。其中，Ring 0级别具有最高权限，可以执行任何指令而没有限制。运行级别从Ring 0到Ring 3依次递减。应用程序一般运行在Ring 3级别。操作系统内核态代码运行在Ring 0级别，因为它需要直接控制和修改CPU的状态，而类似这样的操作需要运行在Ring0级别的特权指令才能完成。

在x86体系结构中实现虚拟化，需要在客户操作系统层以下加入虚拟化

层，来实现物理资源的共享。可见，这个虚拟化层运行在Ring 0级别，而客户操作系统只能运行在Ring 0以上的级别，如图2.5所示。

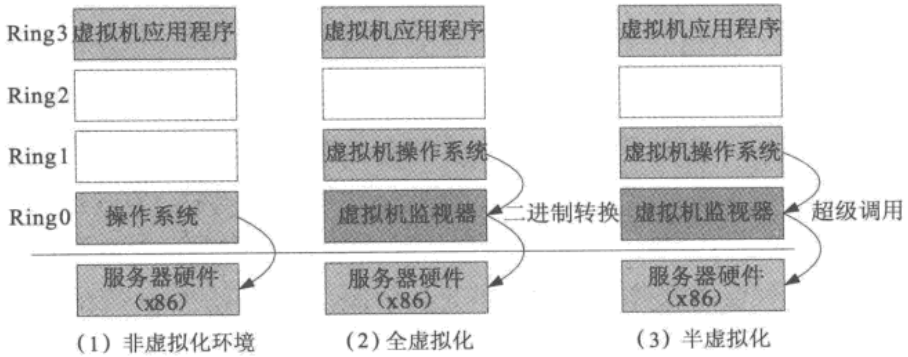



图2.5　x86体系结构下的软件CPU虚拟化

但是，客户操作系统中的特权指令，如中断处理和内存管理指令，如果不运行在Ring 0级别将会具有不同的语义，产生不同的效果，或者根本不产生作用。由于这些指令的存在，使虚拟化x86体系结构并不那么轻而易举。问题的关键在于这些在虚拟机里执行的敏感指令不能直接作用于真实硬件之上，而需要被虚拟机监视器接管和模拟。

目前，为了解决x86体系结构下的CPU虚拟化问题，业界提出了全虚拟化（Full-virtualization）和半虚拟化（Para-virtualization）两种不同的软件方案，如图2.5所示。除了通过软件的方式实现CPU虚拟化外，业界还提出了在硬件层添加支持功能的硬件辅助虚拟化（Hardware Assisted Virtualization）方案来处理这些敏感的高级别指令。

全虚拟化采用二进制代码动态翻译技术（Dynamic Binary Translation）来解决客户操作系统的特权指令问题，如图2.5（2）所示。所谓二进制代码动态翻译，是指在虚拟机运行时，在敏感指令前插入陷入指令，将执行陷入到虚拟机监视器中。虚拟机监视器会将这些指令动态转换成可完成相同功能的指令序列后再执行。通过这种方式，全虚拟化将在客户操作系统内核态执行的敏感指令转换成可以通过虚拟机监视器执行的具有相同效果的指令序列，而对于非敏感指令则可以直接在物理处理器上运行。形象地说，在全虚拟化中，虚拟机监视器在关键的时候“欺骗”虚拟机，使得客户操作系统还以为自己在真实的物理环境下运行。全虚拟化的优点在于代码的转换工作是动态完成的，无需修改客户操作系统，因而可以支持多种操作系统。然而，全虚拟化中的动态转换需要一定的性能开销。Microsoft Virtual PC、Microsoft Virtual Server、VMware WorkStation和VMware ESX Server的早期版本都采用全虚拟化技术。

与全虚拟化不同，半虚拟化通过修改客户操作系统来解决虚拟机执行特权指令的问题。在半虚拟化中，被虚拟化平台托管的客户操作系统需要修改其操作系统，将所有敏感指令替换为对底层虚拟化平台的超级调用（Hypercall），如图2.5（3）所示。虚拟化平台也为这些敏感的特权指令提供了调用接口。形象地说，半虚拟化中的客户操作系统被修改后，知道自己处在虚拟化环境中，从而主动配合虚拟机监视器，在需要的时候对虚拟化平台进行调用来完成敏感指令的执行。在半虚拟化中，客户操作系统和虚拟化平台必须兼容，否则虚拟机无法有效地操作宿主物理机，所以半虚拟化对不同版本操作系统的支持有所限制。Citrix的Xen、VMware 的ESX Server和Microsoft的Hyper—V的最新版本都采用了半虚拟化技术。

无论是全虚拟化还是半虚拟化，它们都是纯软件的CPU虚拟化，不要求对x86架构下的处理器本身进行任何改变。但是，纯软件的虚拟化解决方案存在很多限制。不论是全虚拟化的二进制翻译技术，还是半虚拟化的超级调用技术，这些中间环节必然会增加系统的复杂性和性能开销。此外，在半虚拟化中，对客户操作系统的支持受到虚拟化平台的能力限制。

由此，硬件辅助虚拟化应运而生。这项技术是一种硬件方案，支持虚拟化技术的CPU加入了新的指令集和处理器运行模式来完成与CPU虚拟化相关的功能。目前，Intel公司和AMD公司分别推出了硬件辅助虚拟化技术Intel VT和AMD-V，并逐步集成到最新推出的微处理器产品中。以Intel VT技术为例，支持硬件辅助虚拟化的处理器增加了一套名为虚拟机扩展（Virtual Machine Extensions，VMX）的指令集，该指令集包括十条左右的新增指令来支持与虚拟化相关的操作。此外，Intel VT为处理器定义了两种运行模式，根模式（root）和非根模式（non-root）。虚拟化平台运行在根模式，客户操作系统运行在非根模式。由于硬件辅助虚拟化支持客户操作系统直接在其上运行，无需进行二进制翻译或超级调用，因此减少了相关的性能开销，简化了虚拟化平台的设计。目前，主流的虚拟化软件厂商也在通过和CPU厂商的合作来提高他们虚拟化产品的性能和兼容性。

## 2. 内存虚拟化

内存虚拟化技术把物理机的真实物理内存统一管理，包装成多个虚拟的物理内存分别供若干个虚拟机使用，使得每个虚拟机拥有各自独立的内存空间。在服务器虚拟化技术中，因为内存是虚拟机最频繁访问的设备，因此内存虚拟化与CPU虚拟化具有同等重要的地位。

在内存虚拟化中，虚拟机监视器要能够管理物理机上的内存，并按每个

虚拟机对内存的需求划分机器内存，同时保持各个虚拟机对内存访问的相互隔离。从本质上讲，物理机的内存是一段连续的地址空间，上层应用对于内存的访问多是随机的，因此虚拟机监视器需要维护物理机里内存地址块和虚拟机内部看到的连续内存块的映射关系，保证虚拟机的内存访问是连续的、一致的。现代操作系统中对于内存管理采用了段式、页式、段页式、多级页表、缓存、虚拟内存等多种复杂的技术，虚拟机监视器必须能够支持这些技术，使它们在虚拟机环境下仍然有效，并保证较高的性能。

在讨论内存虚拟化之前，我们先回顾一下经典的内存管理技术。内存作为一种存储设备是程序运行所必不可少的，因为所有的程序都要通过内存将代码和数据提交到CPU进行处理和执行。如果计算机中运行的应用程序过多，就会耗尽系统中的内存，成为提高计算机性能的瓶颈。之前，人们通常利用扩展内存和优化程序来解决该问题，但是该方法成本很高。因此，虚拟内存技术诞生了。为了虚拟内存，现在所有基于x86架构的CPU都配置了内存管理单元（Memory Management Unit，MMU）和页表转换缓冲（Translation Lookaside Buffer，TLB），通过它们来优化虚拟内存的性能。总之，经典的内存管理维护了应用程序所看到的虚拟内存和物理内存的映射关系。

为了在物理服务器上能够运行多个虚拟机，虚拟机监视器必须具备管理虚拟机内存的机制，也就是具有虚拟机内存管理单元。由于新增了一个内存管理层，所以虚拟机内存管理与经典的内存管理有所区别。虚拟机中操作系统看到的“物理”内存不再是真正的物理内存，而是被虚拟机监视器管理的“伪”物理内存。与这个“物理”内存相对应的是新引入的概念——机器内存。机器内存是指物理服务器硬件上真正的内存。在内存虚拟化中存在着逻辑内存、“物理”内存和机器内存三种内存类型，如图2.6所示。而这三种内存的地址空间被称为逻辑地址、“物理”地址和机器地址。

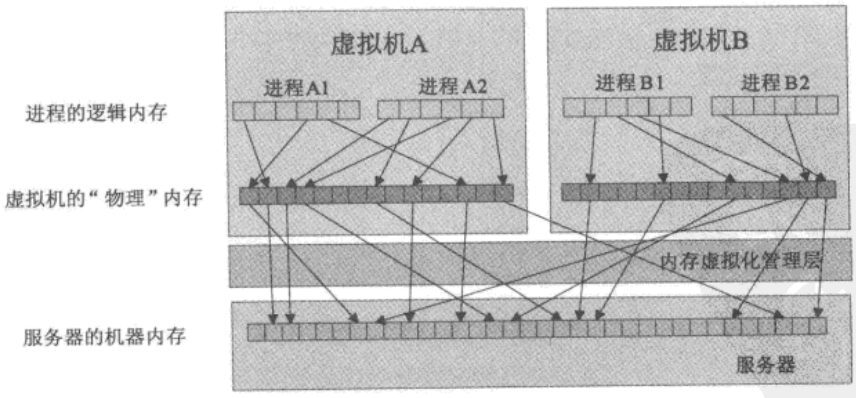
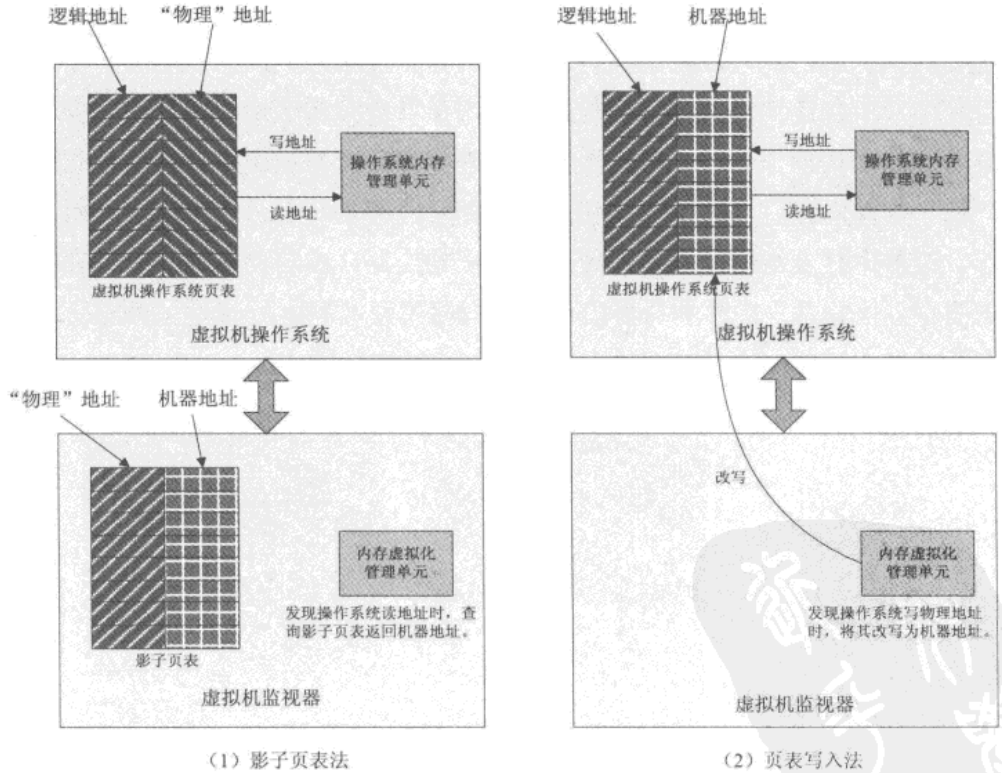



图2.6　内存虚拟化

在内存虚拟化中，逻辑内存与机器内存之间的映射关系是由内存虚拟化管理单元来负责的。内存虚拟化管理单元的实现主要有两种方法。

第一种是影子页表法，如图2.7（1）所示。客户操作系统维护着自己的页表，该页表中的内存地址是客户操作系统看到的“物理”地址。同时，虚拟机监视器也为每台虚拟机维护着一个对应的页表，只不过这个页表中记录的是真实的机器内存地址。虚拟机监视器中的页表是以客户操作系统维护的页表为蓝本建立起来的，并且会随着客户操作系统页表的更新而更新，就像它的影子一样，所以被称为“影子页表”。VMware Workstation、VMware ESX Server和KVM都采用了影子页表技术。

第二种是页表写入法，如图2.7（2）所示。当客户操作系统创建一个新页表时，需要向虚拟机监视器注册该页表。此时，虚拟机监视器将剥夺客户操作系统对页表的写权限，并向该页表写入由虚拟机监视器维护的机器内存地址。当客户操作系统访问内存时，它可以在自己的页表中获得真实的机器内存地址。客户操作系统对页表的每次修改都会陷入虚拟机监视器，由虚拟机监视器来更新页表，保证其页表项记录的始终是真实的机器地址。页表写入法需要修改客户操作系统，Xen是采用该方法的典型代表。




（1）影子页表法　　（2）页表写入法

图2.7　内存虚拟化的两种方法

### 3. 设备与I/O虚拟化

除了处理器与内存外，服务器中其他需要虚拟化的关键部件还包括设备与I/O。设备与I/O虚拟化技术把物理机的真实设备统一管理，包装成多个虚拟设备给若干个虚拟机使用，响应每个虚拟机的设备访问请求和I/O请求。

目前，主流的设备与I/O虚拟化都是通过软件的方式实现的。虚拟化平台作为在共享硬件与虚拟机之间的平台，为设备与I/O的管理提供了便利，也为虚拟机提供了丰富的虚拟设备功能。

以VMware的虚拟化平台为例，虚拟化平台将物理机的设备虚拟化，把这些设备标准化为一系列虚拟设备，为虚拟机提供一个可以使用的虚拟设备集合，如图2.8所示。值得注意的是，经过虚拟化的设备并不一定与物理设备的型号、配置、参数等完全相符，然而这些虚拟设备能够有效地模拟物理设备的动作，将虚拟机的设备操作转译给物理设备，并将物理设备的运行结果返回给虚拟机。这种将虚拟设备统一并标准化的方式带来的另一个好处就是虚拟机并不依赖于底层物理设备的实现。因为对于虚拟机来说，它看到的始终是由虚拟化平台提供的这些标准设备。这样，只要虚拟化平台始终保持一致，虚拟机就可以在不同的物理平台上进行迁移。

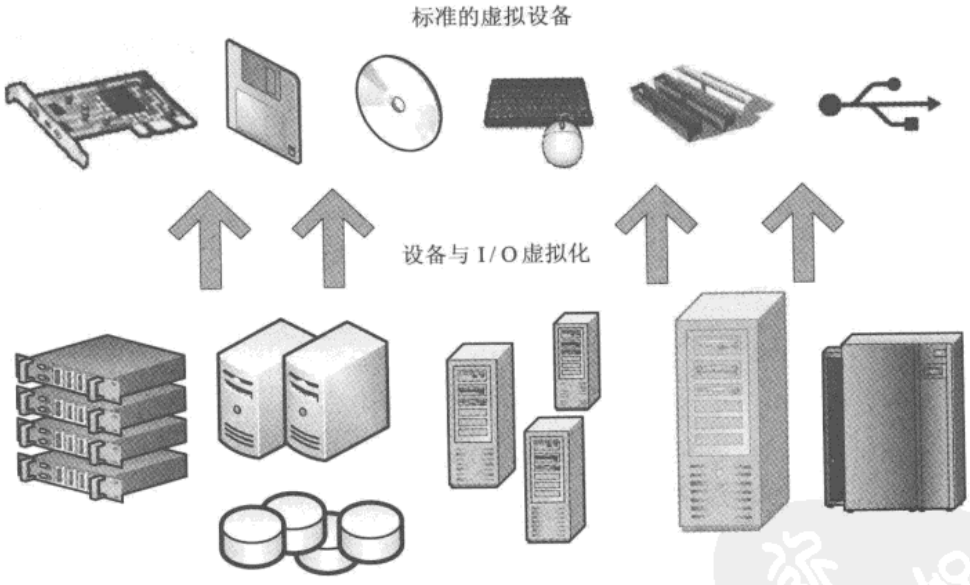



图2.8　设备与I/O虚拟化

在服务器虚拟化中，网络接口是一个特殊的设备，具有重要的作用。虚拟服务器都是通过网络向外界提供服务的。在服务器虚拟化中每一个虚

拟机都变成了一个独立的逻辑服务器，它们之间的通信通过网络接口进行。每一个虚拟机都被分配了一个虚拟的网络接口，从虚拟机内部看来就是一块虚拟网卡。服务器虚拟化要求对宿主操作系统的网络接口驱动进行修改。经过修改后，物理机的网络接口不仅要承担原有网卡的功能，还要通过软件虚拟出一个交换机，如图2.9所示。虚拟交换机工作于数据链路层，负责转发从物理机外部网络投递到虚拟机网络接口的数据包，并维护多个虚拟机网络接口之间的连接。当一个虚拟机与同一个物理机上的其他虚拟机通信时，它的数据包会通过自己的虚拟网络接口发出，虚拟交换机收到该数据包后将其转发给目标虚拟机的虚拟网络接口。这个转发过程不需要占用物理带宽，因为有虚拟化平台以软件的方式管理着这个网络。

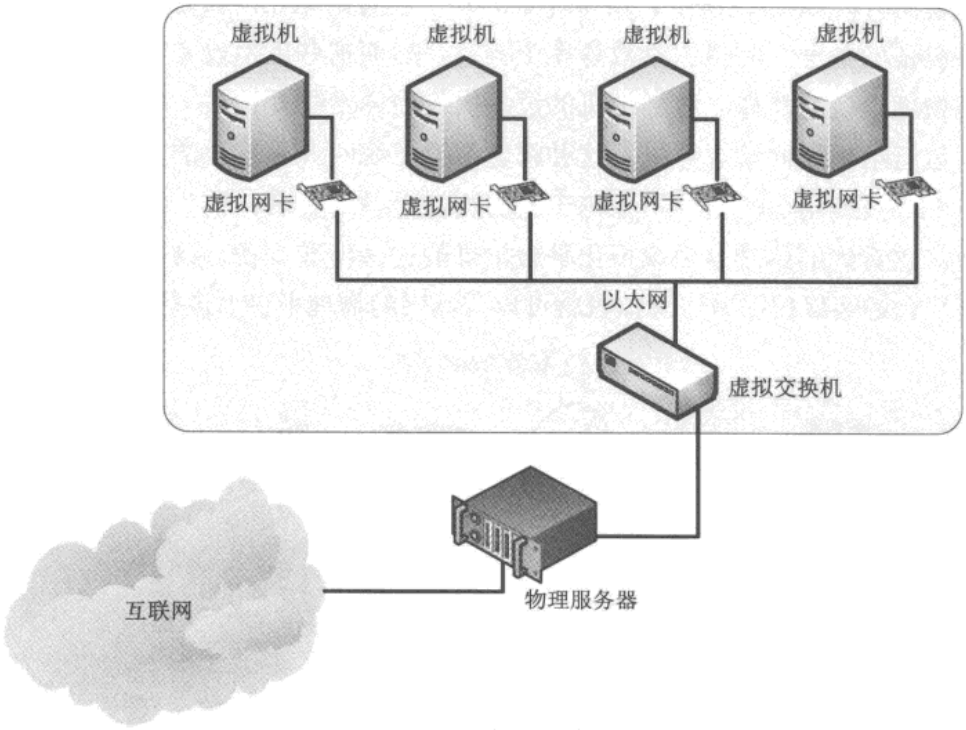



图2.9　网络接口虚拟化

### 4. 实时迁移技术

实时迁移（Live Migration）技术是在虚拟机运行过程中，将整个虚拟机的运行状态完整、快速地从原来所在的宿主机硬件平台迁移到新的宿主机硬件平台上，并且整个迁移过程是平滑的，用户几乎不会察觉到任何差异，如图2.10所示。由于虚拟化抽象了真实的物理资源，因此可以支持原宿主机和目标宿主机硬件平台的异构性。

实时迁移需要虚拟机监视器的协助，即通过源主机和目标主机上虚拟机监视器的相互配合，来完成客户操作系统的内存和其他状态信息的拷贝。实时迁移开始以后，内存页面被不断地从源虚拟机监视器拷贝到目标虚拟机监视器。这个拷贝过程对源虚拟机的运行不会产生影响。最后一部分内存页面被拷贝到目标虚拟机监视器之后，目标虚拟机开始运行，虚拟机监视器切换源虚拟机与目标虚拟机，源虚拟机的运行被终止，实时迁移过程完成。

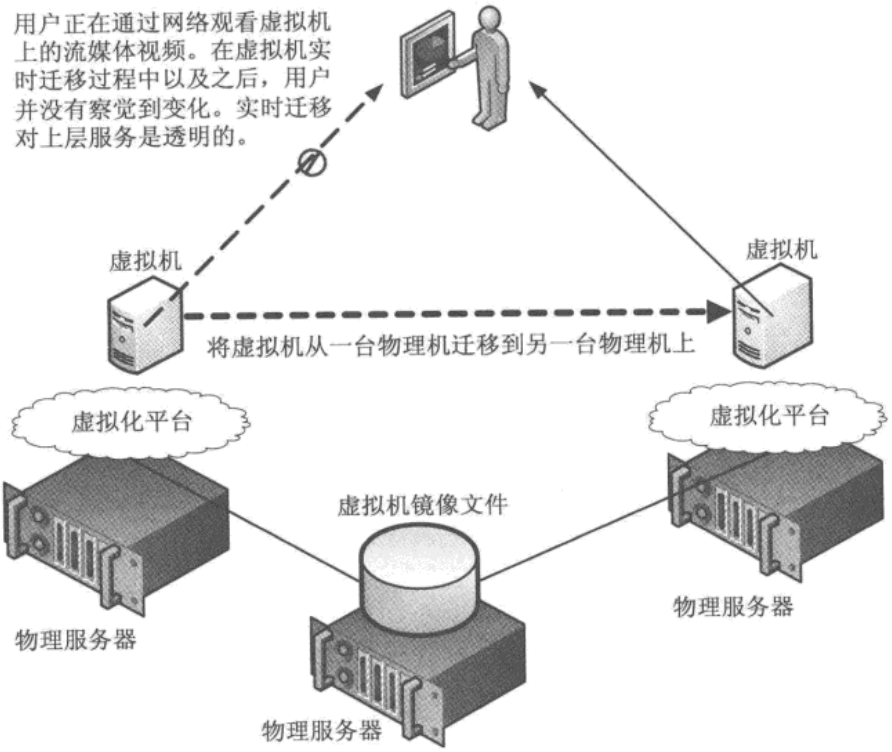



图2.10　实时迁移技术示意图

实时迁移技术最初只应用在系统硬件维护方面。众所周知，数据中心的硬件需要定期地进行维护和更新，而虚拟机上的服务需要7×24不间断地运行。如果使用实时迁移技术，便可以在不宕机的情况下，将虚拟机迁移到另外一台物理机上，然后对原来虚拟机所在的物理机进行硬件维护。维护完成以后，虚拟机迁回到原来的物理机上，整个过程对用户是透明的。目前，实时迁移技术更多地被用做资源整合，通过优化的虚拟机动态调度方法，数据中心的资源利用率可以得到进一步提升。

### 2.2.5 性能分析

服务器虚拟化的性能一直是人们所关注的问题。一方面，采用服务器

虚拟化技术以后，虚拟服务器上的应用与直接运行在物理服务器上的应用相比性能是否有很大差异；另一方面，服务器虚拟化的不同实现技术所提供的性能是否有很大差异。

首先，我们从应用对资源的利用情况进行服务器虚拟化的性能分析，大致可以把应用分为三种类型：处理器密集型（CPU Intensive）、内存密集型（Memory Intensive）和输入/输出密集型（I/O Intensive）。

对于处理器密集型应用，它们需要消耗大量处理器资源，使得处理器保持一个较高的利用率，而处理器的调度是由物理服务器的操作系统内核或虚拟化平台的内核管理的。在物理服务器上，操作系统直接对应用的进程进行调度；在虚拟化平台上，操作系统直接对虚拟机的进程进行调度，并间接地影响虚拟机内部应用的进程，引入了调度开销。对于不同的虚拟化平台，实现处理器调度的机制和策略不同，开销的大小也有差异。

对于内存密集型应用，它们需要频繁使用内存空间，而物理内存和虚拟内存的映射和读写操作也是由物理服务器的操作系统内核或虚拟化平台的内核管理的。在物理服务器上，内存管理单元直接负责虚拟内存和物理内存的寻址；而在虚拟化平台下，虚拟机操作系统所管理的是虚拟内存和伪“物理”内存间的映射，虚拟化平台的内存管理单元管理着伪“物理”内存和真正的机器内存之间的映射，增加的这层映射关系造成了内存寻址的开销。各种虚拟化平台所采用的内存寻址机制也有差别，导致了性能的不同。

对于输入/输出密集型应用，它们需要通过网络和外界进行频繁的通信。在物理服务器上，操作系统的网络驱动直接作用于物理网卡上，因此，应用能够直接通过网络驱动和物理网卡与外界进行通信。而虚拟化平台为每个虚拟机创建的是虚拟网卡，这些虚拟网卡分时共享真正的物理网卡，应用在网络通信过程中，数据包会在虚拟网卡到物理网卡之间进行分发和转换，造成了一定的开销。

VMware公司曾经公布过一份服务器虚拟化（全虚拟化）的性能报告，它评估了上述三类应用分别在物理服务器、VMware ESX v3.01GA和Xen v3.03-0上运行的性能。在这份报告公布后不久，Xen公司也发布了一份类似的报告，不过它评估的是应用在物理服务器、Xen Enterprise v3.2（公共测试版）和VMware ESX v3.0.1 GA上的性能。这两份报告验证了虚拟服务器与物理间的性能差异并不明显，两款不同的虚拟化软件则是各有千秋。

为了测试处理器密集型应用在物理服务器和虚拟化平台上的性能，这

两份报告都选取了标准测试工具SPECcpu2000 Integer。从测试结果来看，处理器密集型应用运行在物理服务器（Native）和虚拟化平台（VMware ESX, Xen/Xen Enterprise）上的性能的差异很小（低于5%），就虚拟化平台而言，VMware ESX的表现略优于Xen/Xen Enterprise。为了测试内存密集型应用在物理服务器和虚拟化平台上的性能，这两份报告都选取了标准测试工具Passmark。从测试结果来看，内存密集型应用运行在物理服务器（Native）和虚拟化平台（VMware ESX, Xen/Xen Enterprise）上的性能差异也很小（低于5%），就虚拟化平台而言，VMware ESX v3.0.1优于Xen v3.03—0，而经过改进的Xen Enterprise v3.2优于VMware ESX v3.0.1。为了测试输入/输出密集型应用在物理服务器和虚拟化平台上的性能，这两份报告都选取了标准测试工具Netperf。从测试结果来看，输入/输出密集型应用运行在物理服务器、VMware ESX v3.0.1和Xen Enerprise v3.2的性能较为接近。

除了对不同类型应用的评估，我们也可以从服务质量的维度来评估服务器虚拟化的性能，衡量Web服务的两个重要指标是吞吐量（Throughput）和响应时间(Response Time)。相同条件下，吞吐量越大，说明服务同时处理请求的能力越强、响应时间越短，也就是说，服务处理单个事务的速度越快。

衡量标准的处理器密集型应用、内存密集型应用和输入/输出密集型应用的性能对实际应用具有很好的参考价值。但在现实场景中运行的往往是具有各种业务逻辑的应用，例如典型的J2EE应用，它不同层次上的功能部件对于资源的需求是不一样的。因此，衡量一个具体类型的商务应用的综合性能往往对企业构建虚拟化环境具有更大的指导意义。IBM和VMware公司曾联合评估过企业级J2EE应用服务器WebSphere Application Server（WAS）v7在VMware ESX v3.5虚拟环境下的性能，它所采用的标准测试工具是模拟股票交易系统DayTrader v1.2。对于运行在VMware ESX v3.5上的WAS独立应用（WAS Standalone），随着分配给它的虚拟CPU数量的增加，其吞吐量也相应增加，相对于直接运行在物理服务器上，它的吞吐量有10%以内的下降。但是，如果多台虚拟机组成的WAS集群运行在同一个配有单个多核处理器的虚拟化平台上，情况则有所不同：在默认配置下，随着分配给每个虚拟机的虚拟CPU数量的增加，吞吐量也相应增加，当分配的虚拟CPU等于和多于4个时，吞吐量甚至超过了直接运行在物理服务器上所对应的吞吐量。这说明ESX所采用的调度策略考虑了CPU多核之间的亲和性（Affinity），避免了虚拟机进程在各个核之间频繁迁移而造成损

耗，而物理服务器上的普通操作系统进程调度策略并没有考虑到这一点。

除了横向比较x86架构下的服务器虚拟化性能，比较大型机虚拟化平台（z/ VM）和x86虚拟化平台的性能也具有现实意义。这样的测试在分配了相似的物理资源的条件下（8Cores@4GHz）不断地增加运行在z/VM和x86上的虚拟机数量，通过标准测试工具来测试吞吐量、响应时间和CPU利用率。响应时间的测试结果显示，当虚拟机数量超过20时，x86虚拟化平台上虚拟机的响应时间会迅速增加，直至达到其容纳虚拟机的极限（约50个）；而z/ VM则表现出良好的性能，即使在虚拟机数量达到100个时，响应时间也只是微量增长。在吞吐量的测试中，随着虚拟机数量的增加，x86虚拟化平台的吞吐量也随之增加，当虚拟机数量在25～50个时，吞吐量基本维持在每秒50个事务；而z/ VM随着虚拟机数量增加，吞吐量呈对数型增长，最大能达到每秒150个事务的处理能力。这两者的差异是由大型机和x86硬件体系结构不同造成的，大型机在设计之初就考虑到了虚拟化和并行处理等因素，从而充分利用大型机上的资源，而x86只是面向普通的个人用户，一开始并没有考虑支持虚拟化，之后只能以补丁式的方式实现虚拟化，因此所表现出来的性能和大型机是无法比拟的。

总之，通过这些服务器虚拟化的性能测试报告，可以得出以下结论：第一，服务器虚拟化会引入一定的系统开销，应用的性能比直接运行在物理服务器上有所下降，但是随着该技术的日益成熟，以及硬件辅助虚拟化和多核等技术的不断成熟，这个开销已经在逐渐缩小，性能下降的幅度变得可以接受；第二，服务器虚拟化的各种实现技术之间存在一些不同点，但是同等系统架构（如x86）的虚拟化平台的实现方法正在逐步趋同，不同品牌虚拟化平台的性能差异已经很小；第三，大型机的服务器虚拟化技术相比x86的服务器虚拟化技术具有明显的优势，具有更好的服务器整合能力，并使得应用拥有更快的响应时间和更大的吞吐量；第四，对于需要运行在虚拟化环境的企业应用，都应针对其应用的特点进行实际测试调优后才可以上线，从而更好地满足用户对于服务质量的需求。





## 2.2.6 技术优势

### 1. 降低运营成本

企业的数据中心需要持续不断的投资来更新和维护IT基础设施。按照

公认的财务计算方法，企业的IT成本分为两部分，一部分是采购成本，包括购买设备、软件、许可证、服务等，记做资本支出；另一部分是运行和维护成本，包括对IT基础设施的维护和管理等，记做运营支出。

服务器虚拟化使得系统管理员摆脱大量繁重的与物理服务器、操作系统、中间件及兼容性问题打交道的管理工作，更加专注于应用的管理。同时，各大服务器虚拟化厂商都提供了功能强大的虚拟化环境管理工具，降低管理员进行人工干预的频率，并提供更简便、更强大的管理界面。因此，服务器虚拟化可以降低IT基础设施的运营成本，促进企业进一步采用信息化工具和服务。

### 2. 提高应用兼容性

在现有的数据中心中，大量的应用运行在各种互不兼容的环境中，兼容性问题非常突出。开发应用需要考虑硬件平台、操作系统、中间件等各个级别，各种互不兼容的应用也大大增加了管理、维护和整合的难度。

服务器虚拟化技术提供的封装和隔离特性使得应用所在的平台与底层服务器环境隔离，管理员不再需要根据底层环境的变化频繁地调整应用，仅需构建一个应用版本，并将其发布到被虚拟化封装后的不同类型的平台上。

### 3. 加速应用部署

在传统的数据中心中，部署一个应用需要以下几个步骤：寻找合适的物理机、安装操作系统、安装中间件、安装应用、配置、测试、运行。安装一个应用通常需要耗费十几个小时甚至几天，并且需要部署人员全程跟踪部署进度，执行下一步操作。这样的部署方式很容易出现错误，例如安装被错误中止、不同领域不同模块的安装部署人员在沟通或交接时出现差错等。

采用服务器虚拟化以后，部署一个应用其实就是部署一个封装好的操作系统和应用程序的虚拟机，部署过程只需要以下几个步骤：输入激活配置参数、拷贝虚拟机、启动虚拟机、激活（配置）虚拟机。通常这样的部署只需要几分钟至几十分钟，相对于传统的应用部署方式，不需要人工干预，缩短了部署时间，降低了部署成本。

### 4. 提高服务可用性

服务可用性是指服务能够持续、可靠地运行的能力。服务的高可用性

要求将日常维护操作对服务的影响降到最低，即便发生系统故障或硬件失效，服务也可以在较短的时间内被恢复。传统的数据中心为了保证服务可用性，需要采用一定的措施，如采用多机备份、冗余等技术，并通过额外的可用性管理工具来监控、调度和管理服务。

采用了服务器虚拟化技术以后，服务可用性得到了有效提升，而且易于实现。在采用了虚拟化的数据中心里，由于虚拟机是单个的逻辑文件，并且对应的处理器和内存资源都被虚拟机管理程序封装和隔离，因此用户可以方便地对运行中的虚拟机快照并备份成虚拟机镜像文件。在需要的时候动态迁移虚拟机，将它恢复到某个备份，或者在其他物理机上运行该备份以提高可用性。这样，用户可以得到更高的服务可用性。

### 5. 提升资源利用率

根据Gartner的调查研究报告，在当前的企业数据中心中，出于管理简便、安全性和性能的考虑，绝大多数x86服务器上都只运行一个应用，导致服务器的CPU利用率普遍偏低，平均只有5%～20%。

采用服务器虚拟化技术，数据中心管理员可以将原有的多台服务器整合到一台物理服务器上，提高物理服务器的使用率，同时通过虚拟化技术提供的隔离性、封装性，保证原有服务仍然可用，其安全性、性能不会受到影响。据分析，通过对服务器进行虚拟化整合，不仅服务器的CPU利用率得到了提高，而且服务器的内存利用率、存储利用率和网络利用率也得到了大幅度提高。

### 6. 动态调度资源

服务器虚拟化技术的关键功能之一是实时迁移。目前各主流虚拟化平台都提供了实时迁移功能。实时迁移可以在不中断服务的情况下将虚拟机从一台物理服务器迁移到另一台物理服务器。对于数据中心管理员来说，看到的数据中心不再是一台台隔离的服务器，而是一个统一的资源池，管理着大量的CPU、内存、存储空间、网络资源，有了实时迁移技术，每个虚拟机可以在池内自由地移动。

同时，服务器虚拟化技术还使得用户可以即时地调整虚拟机的资源，如CPU、内存等，而不是像原来的物理服务器那样需要关闭服务器，打开机箱安装设备，再重新启动系统。虚拟化产品都提供了可以被程序调用的资源调整API，以及用户可操作的界面，这样数据中心管理程序和数据中

心管理员都可以灵活地根据虚拟机内部的资源使用情况灵活调整分配给虚拟机的资源。

### 7. 降低能源消耗

数据中心的计算能耗通常只占总能耗的很小一部分。对于一个标准的x86服务器，一个处理器每小时的能耗一般来说只有几瓦特，而支撑这个服务器运行的总耗电量，如制冷、通风等，则可以达到几百甚至上千瓦特。数据中心的能源消耗问题不可忽视。

关闭利用率不高的服务器是最直观的节能减排方式。在传统模式中，一个应用运行在一台服务器之上，关闭服务器就等于关闭了应用。服务器虚拟化为解除应用与物理服务器的绑定提供了可能，在负载低谷时，管理员可以将原来运行在各个服务器上的应用整合到较少的几台服务器上，关闭空闲的物理服务器，通过减少运行的物理服务器数量，减少CPU以外各单元的耗电量，达到绿色节能的目的。

## 2.3 其他虚拟化技术

正如上文所述，本书主要讨论的是在数据中心实施虚拟化，围绕这一主题，服务器虚拟化是我们讨论的重点。除此之外，一个完整的数据中心离不开网络和存储等基础设施。同样的，在交付应用时，软件虚拟化也会为数据中心的应用管理提供极大的便利。因此，在本节我们将介绍在数据中心中其他几个较为重要的虚拟化技术，它们是网络虚拟化、存储虚拟化、应用虚拟化和桌面虚拟化。

### 2.3.1 网络虚拟化

网络虚拟化通常包括虚拟局域网和虚拟专用网。虚拟局域网可以将一个物理局域网划分成多个虚拟局域网，甚至将多个物理局域网里的节点划分到一个虚拟的局域网中，使得虚拟局域网中的通信类似于物理局域网的方式，并对用户透明。虚拟专用网对网络连接进行了抽象，允许远程用户访问组织内部的网络，就像物理上连接到该网络一样。虚拟专用网帮助管理员保护IT环境，防止来自Internet或Intranet中不相干网段的威胁，同时使用户能够快速、安全地访问应用程序和数据。目前虚拟专用网在大量的办公环境中都有使用，成为移动办公的一个重要支撑技术。

最近，各厂商又为网络虚拟化技术增添了新的内容。对于网络设备提供商来说，网络虚拟化是对网络设备的虚拟化，即对传统的路由器、交换机等设备进行增强，使其可以支持大量的可扩展的应用，同一网络设备可以运行多个虚拟的网络设备，如防火墙、VoIP、移动业务等。

目前网络虚拟化还处于初级阶段，有大量的基础问题需要解决，比如更复杂的网络通信，识别物理与虚拟网络设备等。

### 2.3.2 存储虚拟化

随着信息业务的不断发展，网络存储系统已经成为企业的核心平台，大量高价值数据积淀下来，围绕这些数据的应用对平台的要求也越来越高，不仅是在存储容量上，还包括数据访问性能、数据传输性能、数据管理能力、存储扩展能力等多个方面。可以说，存储网络平台的综合性能的优劣，将直接影响到整个系统的正常运行。正因为这个原因，虚拟化技术又一子领域——存储虚拟化技术应运而生。

RAID（Redundant Array of Independent Disk）技术是存储虚拟化技术的雏形。它通过将多块物理磁盘以阵列的方式组合起来，为上层提供一个统一的存储空间。对操作系统及上层的用户来说，他们并不知道服务器中有多少块磁盘，只能看到一块大的“虚拟”的磁盘，即一个逻辑存储单元。在RAID技术之后出现的是NAS（Network Attached Storage）和SAN（Storage Area Network）。NAS将文件存储与本地计算机系统解耦合，把文件存储集中在连接到网络上的NAS存储单元，如NAS文件服务器。其他网络上的异构设备都可以通过标准的网络文件访问协议，如UNIX系统下的NFS（Network File System）和Window系统下的SMB（Server Message Block），来对其上的文件按照权限限制进行访问和更新。与NAS不同，虽然同样是将存储从本地系统上分离，集中在局域网上供用户共享与使用，SAN一般是由磁盘阵列连接光纤通道组成，服务器和客户机通过SCSI协议进行高速数据通信，SAN用户感觉这些存储资源和直接连接在本地系统上设备是一样的。在SAN中，存储的共享是在磁盘区块的级别上，而在NAS中是在文件级别上。

目前，不限于RAID、NAS和SAN，存储虚拟化被赋予了更多的含义。存储虚拟化可以使逻辑存储单元在广域网范围内整合，并且可以不需要停机就从一个磁盘阵列移动到另一个磁盘阵列上。此外，存储虚拟化还可以

根据用户的实际使用情况来分配存储资源。例如，操作系统磁盘管理器给用户分配了300GB空间，但用户当前使用量只有2GB，而且在一段时间内保持稳定，则实际被分配的空间可能只有10GB，小于提供给用户的标称容量。而当用户实际使用量增加时，再适当分配新的存储空间。这样有利于提升资源利用率。

### 2.3.3 桌面虚拟化

桌面虚拟化将用户的桌面环境与其使用的终端设备解耦合。服务器上存放的是每个用户的完整桌面环境。用户可以使用不同的具有足够处理和显示功能的终端设备，如个人电脑或智能手机等，通过网络访问该桌面环境，如图2.11所示。桌面虚拟化的最大好处就是能够使用软件从集中位置来配置PC及其他客户端设备。系统维护部门可以在数据中心，而不是在每个用户的桌面管理众多的企业客户机，这就减少了现场支持工作，并且加强了对应用软件和补丁管理的控制。

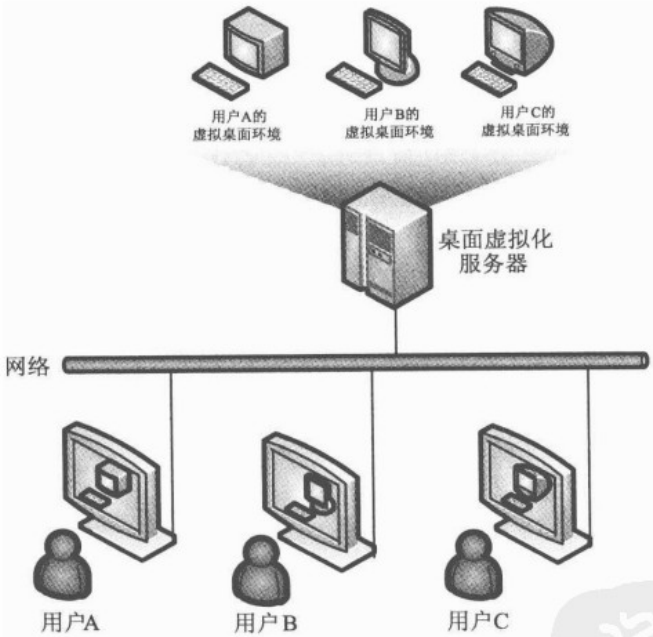



图2.11 桌面虚拟化

桌面虚拟化将众多终端的资源集合到后台数据中心，以便管理者对企业数百上千个终端进行统一认证、统一管理和更为灵活地调配资源。终端用户在实际使用中也不会改变任何使用习惯，通过提供特殊身份认证的智能授权装置，登录任意终端即可获取自身相关数据，继续原有业务，这意

味着灵活性也将大大提高。

不论是桌面虚拟化还是服务器虚拟化，安全是一个不可忽视的问题。在企业内部信息安全中，最危险的元素就是桌面设备，很多企业甚至为此专门推出了桌面终端安全管理软件，以防终端的隐患影响局域网内部其他设备的安全运行和后台重要数据被窃取。而通过桌面虚拟化，所有数据、认证都能做到策略一致、统一管理，有效地提高了企业的信息安全级别。进一步说，通过实施桌面虚拟化，用户可将原有的终端数据资源，甚至操作系统都转移到后台数据中心的服务器中，而前台终端则转化为以显示为主、计算为辅的轻量级客户端。

桌面虚拟化可以协助企业进一步简化轻量级客户端架构。与现有的传统分布式PC桌面系统部署相比，采用桌面虚拟化的轻量级客户端架构部署服务可为企业减少硬件与软件的采购开销，并进一步降低企业的内部管理成本与风险。随着硬件的快速更新换代、应用软件的增加和分布、工作环境的分散，管理和维护终端设备的工作变得越来越困难。桌面虚拟化可以为企业降低电费、管理、PC购买、运行和维护等成本。

桌面虚拟化的另一个好处是，由于用户的桌面环境被保存成一个个虚拟机，通过对虚拟机进行快照、备份，就可以对用户的桌面环境进行快照、备份。当用户的桌面环境被攻击，或者出现重大操作错误时，用户可以恢复保存的备份，这样大大降低了用户和系统管理员的维护负担。





### 2.3.4 应用虚拟化

应用程序在很大程度上依赖于操作系统为其提供的功能，比如内存分配、设备驱动、服务进程、动态链接库等。这些应用程序之间也存在着复杂的依存关系。它们通常共享许多不同的程序部件，比如动态链接库。如果一个程序的正确运行需要一个特定版本的动态链接库，而另一个程序需要这个动态链接库的另一个版本，那么在同一个系统上同时安装这两个应用程序，就会造成动态链接库的冲突，其中一个程序会覆盖另一个程序所需要的动态链接库，造成另一个程序的不可用。因此，系统或其他应用程序的改变（如执行升级补丁等）都有可能导致应用之间的不兼容。当一个企业要为其组织中的桌面系统安装新应用时，总是要进行严格而烦琐的测试，来保证新应用与系统中的已有应用不产生冲突。这个过程需要耗费大量的人力、物力和财力。因为这个原因，虚拟化技术的又一子领域——应

用虚拟化技术应运而生，如图2.12所示。

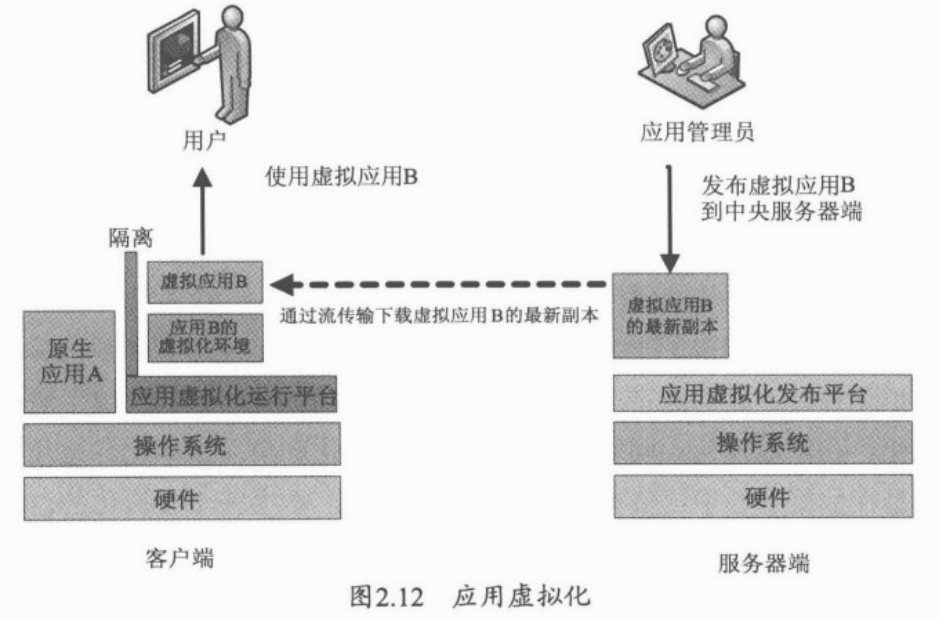



图2.12　应用虚拟化

有了应用虚拟化，应用可以运行在任何共享的计算资源上。应用虚拟化为应用程序提供了一个虚拟的运行环境。在这个环境中，不仅拥有应用程序的可执行文件，还包括它所需要的运行时环境。应用虚拟化为企业内部的IT管理提供了便利。在应用虚拟化以前，如果管理员要对一个应用程序进行更新，他必须处理每一台机器可能出现的不同类型的不兼容情况。采用应用虚拟化技术后，管理员只需要更新虚拟环境中的应用程序副本，并将其发布出去；使用者也与传统的应用程序安装方式不同，程序并不是完全安装在本地机器的硬盘上，而是从一个中央服务器上下载下来，运行在本地的应用虚拟化环境中。当用户关闭应用程序后，已经下载下来的部分可以被完全删除，就像它从来没有在本地机器里运行过一样。

应用虚拟化的应用也可以以流的方式发布到客户端。采用这种方式，仅当用户需要时按需地将程序的部分或者全部内容以流的方式传送到客户端。这种用流方式传送应用程序的方式与用流方式传送多媒体文件的方式有相似之处，要求一定的网络带宽和质量来保证应用在客户端的可用性与易用性。

从本质上说，应用虚拟化是把应用对底层的系统和硬件的依赖抽象出来，从而解除应用与操作系统和硬件的耦合关系。应用程序运行在本地的应用虚拟化环境中，这个环境为应用程序屏蔽了底层可能与其他应用产生冲突的内容，如动态链接库等。这简化了应用程序的部署或升级，因为程

序运行在本地的虚拟环境中，不会与本地安装的其他程序产生冲突，同时带来应用程序升级的便利。

## 2.4 小结

本章概括介绍了虚拟化技术的概念、主要类型、优势、性能和各种类别的虚拟化技术，希望读者通过阅读本章，能够对虚拟化技术有初步的了解。

虚拟化从划分物理资源与逻辑资源的角度为系统管理员、软件开发者、服务提供者创造了丰富的解决方案。但是前提是使用者必须了解不同的虚拟化种类、它们能带来什么功能、具有哪些优势。我们介绍了服务器虚拟化、存储虚拟化、网络虚拟化、应用虚拟化和桌面虚拟化。这几种虚拟化是在数据中心中可实施的重要虚拟化技术，本书关于虚拟化的讨论主要是围绕以上这些技术展开的。在第3章，我们将为读者介绍实施服务器虚拟化的关键技术。在第4章，我们将概述虚拟化技术的最新业界动态。

CHAPTER 3

# 第3章 虚拟化的关键技术

虚拟化技术给数据中心管理带来了诸多优势，它一方面可以提升基础设施利用率，实现运营开销成本最小化；另一方面可以通过整合应用栈和即时应用镜像部署来实现业务管理的高效敏捷。目前，如何在数据中心实施虚拟化和实施中的关键技术成为业界关注的重点。如图3.1所示，实施虚拟化的顺序按照其生命周期可以简单划分为三个重要阶段：创建、部署和管理。本章将逐一介绍各个阶段所涉及的关键技术。


图3.1 虚拟化解决方案生命周期示意图

## 3.1 创建虚拟化解决方案

虚拟化解决方案的创建一般由服务提供商和服务集成商完成。由于虚拟化解决方案是由一系列虚拟镜像或虚拟器件组成的，因此，在这部分我们首先介绍如何创建基本的虚拟镜像，再描述如何创建、组装和发布虚拟器件，然后讨论虚拟器件发布后的镜像管理，最后阐述将物理机环境转换为虚拟机环境的技术。

### 3.1.1 创建基本虚拟镜像

根据第2章给出的定义，虚拟机是指通过虚拟化软件套件模拟的、具有完整硬件功能的、运行在一个隔离环境中的逻辑计算机系统。虚拟机里的操作系统被称为客户操作系统（Guest Operating System，Guest OS），在客户操作系统上可以安装中间件和上层应用程序，从而构成一个完整的软件栈。虚拟镜像是虚拟机的存储实体，它通常是一个或者多个文件，其中包括了虚拟机的配置信息和磁盘数据，还可能包括内存数据。

虚拟镜像的主要使用场景是开发和测试环境：软件开发人员在虚拟机内部对应用进行开发测试，把虚拟镜像作为应用在初始状态或某一中间状态的备份来使用，这样能够在当前的环境发生不可恢复的变更时方便地用虚拟镜像恢复到所需要的状态。

虚拟镜像大致可以分为两类：一类是在虚拟机停机状态下创建的镜像，由于这时的虚拟机内存没有数据需要保存，因此这种镜像只有虚拟机的磁盘数据；另一类是在虚拟机运行过程中做快照所生成的镜像，在这种情况下，虚拟机内存中的数据会被导出到一个文件中，因此这种镜像能够保存虚拟机做快照时的内存状态，在用户重新使用虚拟机时可以立即恢复到进行快照时的状态，不需要进行启动客户操作系统和软件的工作。由于目前使用较广泛的是停机状态下创建的虚拟镜像，因此下文主要讨论这类虚拟镜像。对于快照技术及快照镜像会在3.2.2小节中做介绍。

创建一个最基本的虚拟镜像的流程包括以下三个步骤：创建虚拟机、安装操作系统和关停虚拟机，如图3.2所示。第一步，在虚拟化管理平台上选择虚拟机类型，并设定虚拟硬件参数。参数主要包括虚拟机的CPU数量、内存大小、虚拟磁盘大小、挂载的虚拟光驱及虚拟磁盘等，其中虚拟磁盘的设定要充分考虑到后续安装软件所需空间的实际情况。虚拟化管理平台将依据这些参数创建相应的虚拟机。第二步，选择客户机操作系统并安装，这个过程一般在虚拟化软件套件提供的虚拟机窗口界面上进行，类似于在一台普通的物理机器上安装操作系统。安装客户机操作系统时要遵循“够用即可”的原则，移除不必要的模块、组件和功能，这样既能提高虚拟机运行时的性能，又可以降低虚拟机受攻击的风险。最后一步是关停虚拟机，保存生成的虚拟镜像和配置文件。经过这三个步骤，一个最基本的虚拟镜像就创建完毕了，整个过程一般需要十几分钟左右。

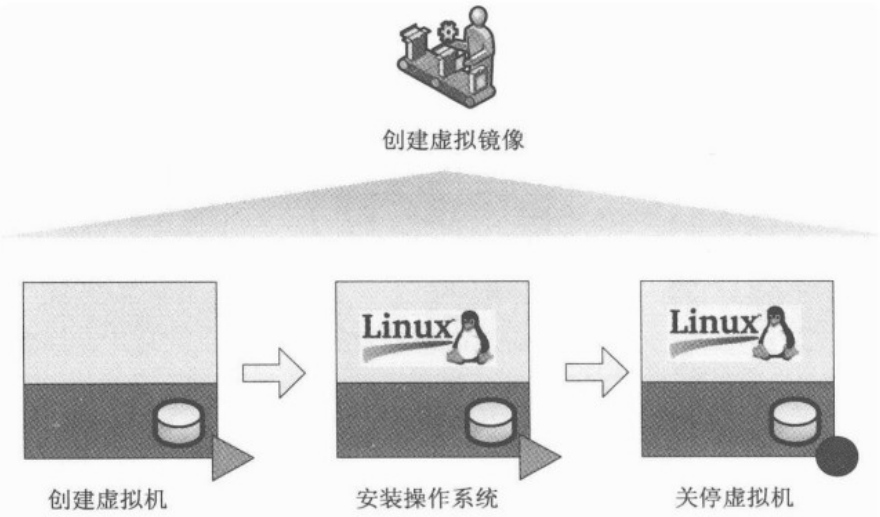


图3.2　创建虚拟镜像流程图

目前主流的虚拟化软件套件都提供了非常方便的虚拟镜像创建功能，一般来说都是图形化、流程化的，用户只需要根据虚拟化软件提供的提示，填写必要的信息，就可以很方便地完成虚拟镜像的创建。

### 3.1.2 创建虚拟器件镜像

在上一节中，我们介绍了如何创建一个最基本的虚拟镜像，但对于用户来说，这样的虚拟镜像并不足以直接使用，因为用户使用虚拟化的目的是希望能够将自己的应用、服务、解决方案运行在虚拟化平台上，而基本虚拟镜像中只安装了操作系统，并没有安装客户需要使用的应用及运行应用所需的中间件等组件。当用户拿到虚拟镜像后，还要进行复杂的中间件安装，以及应用程序的部署和配置工作，加上还需要熟悉虚拟化环境等，反而有可能使用户感觉使用不便了。

虚拟器件（Virtual Appliance）技术能够很好地解决上述难题。虚拟器件技术是服务器虚拟化技术和计算机器件（Appliance）技术结合的产物，有效吸收了两种技术的优点。根据Wikipedia的定义，计算机器件是具有特定功能和有限的配置能力的计算设备，例如硬件防火墙、家用路由器等设备都可以看做是计算机器件。虚拟器件则是一个包括了预安装、预配置的操作系统、中间件和应用的最小化的虚拟机。如图3.3所示，和虚拟镜像相比，虚拟器件文件中既包含客户操作系统，也包含中间件及应用软件，用户拿到虚拟器件文件后经过简单的配置即可使用。与计算机器件相比，虚

拟器件摆脱了硬件的束缚，可以更加容易地创建和发布。

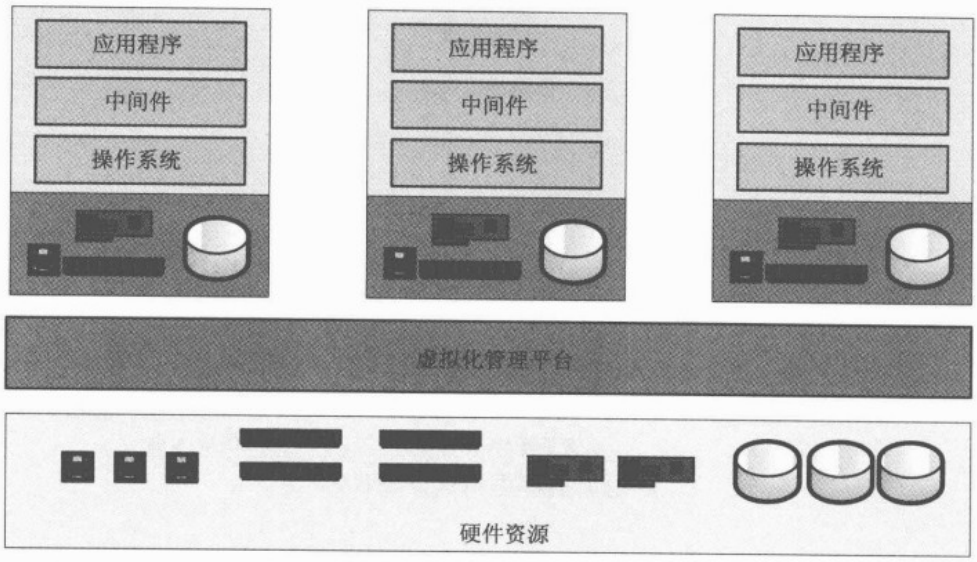



图3.3　虚拟器件结构图

虚拟器件的一个主要使用场景是软件发布。传统的软件发布方式是软件提供商将自己的软件安装文件刻成光盘或者放在网站上，用户通过购买光盘或者下载并购买软件许可证的方法得到安装文件，然后在自己的环境中安装。对于大型的应用软件和中间件，则还需要进行复杂的安装配置，整个过程可能耗时几个小时甚至几天。而采用虚拟器件技术，软件提供商可以将自己的软件及对应的操作系统打包成虚拟器件，供客户下载，客户下载到虚拟器件文件后，在自己的虚拟化环境中启动虚拟器件，再进行一些简单的配置就可以使用，这样的过程只耗时几分钟到几十分钟。可以看出，通过采用虚拟器件的方式，软件发布的过程被大大简化了。认识到虚拟器件的好处之后，很多软件提供商都已经开始采用虚拟器件的方式来发布软件。例如，VMware的官方网站已经有“虚拟器件市场”；在Amazon EC2环境里，虚拟器件已经用于商业目的；IBM的内部网站上包含IBM主要软件产品的虚拟器件正在被大量下载和使用。可以预见，在不远的将来，虚拟器件将成为最为普及的软件和服务的发布方式，用户不再需要花费大量的人力、物力和时间去安装、配置软件，工作效率会得到很大提高。

上文谈到了虚拟器件的基本概念及使用场景，而为了更方便、高效地使用虚拟器件，并让它支持复杂的企业级虚拟化解决方案，创建虚拟器件的过程需要一系列技术的支持，如图3.4所示。在制作虚拟器件之前，需要考虑两方面的关键技术：对多个虚拟器件组成的复杂虚拟化解决方案进行预先规划和通过配置元数据和脚本实现虚拟器件的高度灵活性和模板化。

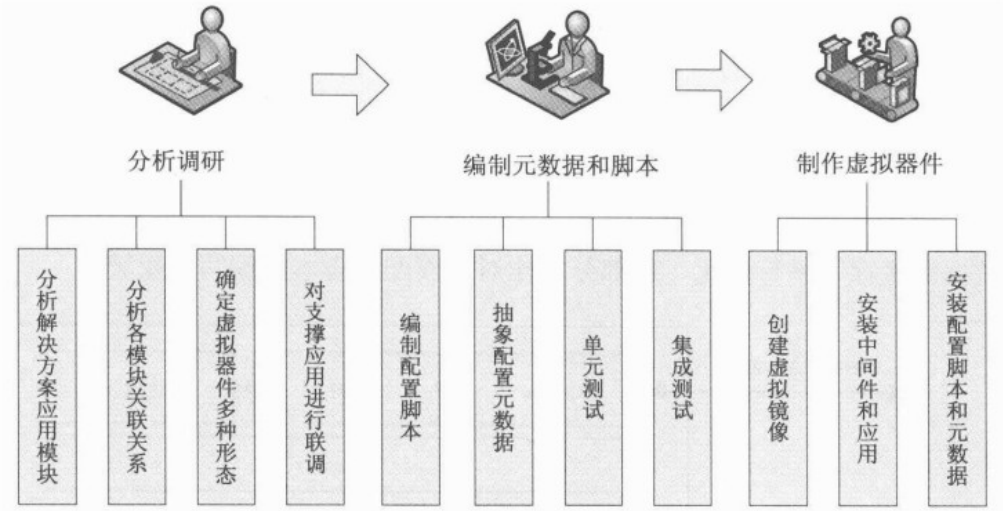


图3.4　虚拟器件创建流程图

虚拟器件在很多场景下都要支持复杂的企业级应用和服务，而应用和服务的特点是需要多个虚拟器件组合交付，在虚拟器件的创建阶段需要考虑各个虚拟器件的关联关系，因而前期调研显得尤为重要。在创建虚拟器件之前，我们首先要调研和分析如何把现有的服务迁移、封装成若干个虚拟器件，然后编写相应配置脚本、规范配置参数并进行多次测试和验证，最后才是真正创建虚拟器件。制作出来的虚拟器件是一个模板，部署者在后续的部署过程中可以将其复制并生成多个实例，将解决方案交付给最终用户。下面详细介绍以上三个阶段的工作。

在开始的调研工作中，需要分析解决方案都由哪些应用模块组成。从基于单机的小型LAMP（Linux-Apache-MySQL-PHP）解决方案（如图3.5所示），到基于集群的企业级解决方案，设计人员需要针对不同的应用场景进行调研工作。例如，IBM公司的模拟股票交易软件Trade，用户虽然只是通过Web方式访问，但是，底层的支撑模块包括了Web服务器（IBM HTTP Server，IHS）、应用服务器集群（IBM WebSphere Application Server，WAS）和后端的数据库（IBM DB2 Server），如图3.6所示，而且，这三者并不是单独运行的实体，它们之间需要相互关联才能支撑模拟股票交易的服务。因此，要将这种复杂的应用封装到多个虚拟器件上，需要对其进行大致的分层或者分类，将不同层次或类型的支撑模块分别安装在不同的虚拟器件中。在前面的例子中，针对于Web服务器、应用服务器和数据库服务器，至少需要三个虚拟器件。需要注意的是，中间件或者应用可能出现多种形态，比如刚才提到的IBM WAS服务器，它可以按需被配置成多种形态，如Deployment Manager、Standalone、Managed Node、Cell等。对于这

种情况，虚拟解决方案中只需要一个WAS虚拟器件就可以了，因为通过在部署阶段读取传入的参数，配置脚本可以将其实例化成上面提到的各种形态。在分层或分类以后，需要考虑支撑模块和操作系统之间的兼容性和配置优化问题。在对支撑模块优化完成以后，还需要对整个解决方案进行联调，目的主要是对网络参数、安全参数等参数进行配置，对请求连接数、数据源缓存等进行优化，这部分工作对后面配置脚本的编写很重要。

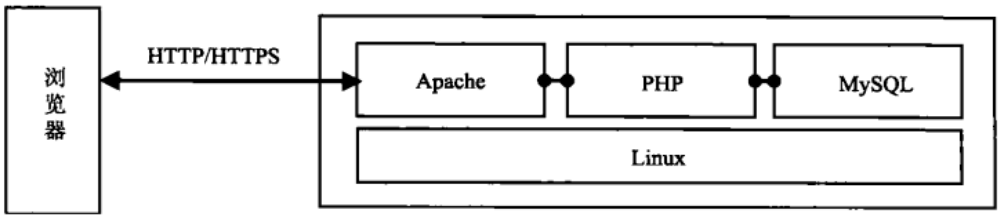



图3.5　LAMP 解决方案

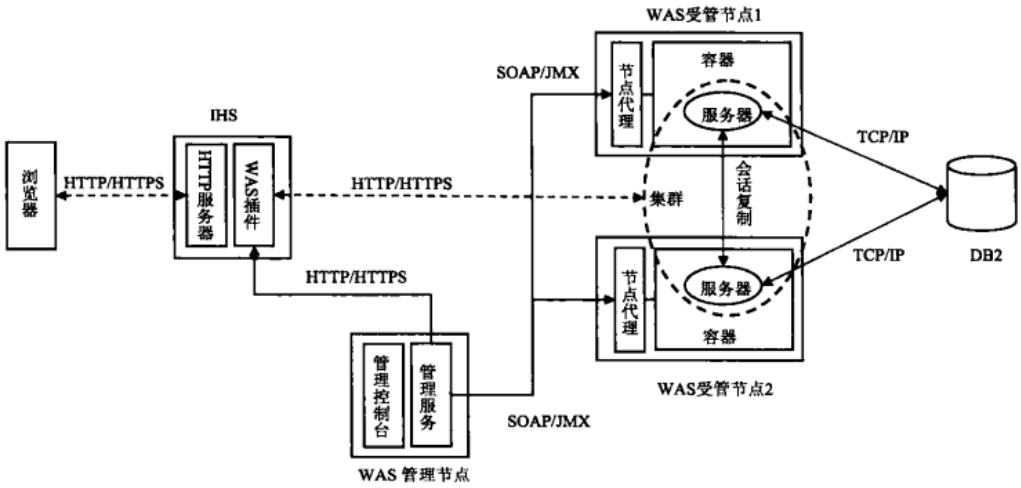



图3.6　IHS-WAS-DB2 解决方案

调研工作完成以后，设计人员就可以编写配置脚本并进行测试了。在前期工作中，我们知道了如何对虚拟器件操作系统和支撑模块调优，由于虚拟器件中的软件栈已经固定，因此这些调优基本上都是一次性的，只需要在创建虚拟器件时配置成最优的固定值即可。但是，对中间件或模块的多态处理、联调时的网络配置、应用参数的设定等操作才是虚拟器件能够适应各种部署环境的根本所在。这些内容的配置需要编制脚本，并根据部署时传入的参数完成。通过脚本实现配置的设定是一个相对简单的操作，只要支撑模块开放命令行接口，脚本就能通过执行一系列命令的方法来使得配置生效。在脚本编制完成以后，设计人员需要确定配置参数及调用脚本的逻辑顺序，并进行测试和验证，使得配置脚本能够满足不同实例化的要求。测试过程分为

单元测试和集成测试，单元测试主要检测单个脚本的正确性，而集成测试模拟脚本执行的顺序来逐一测试脚本，以保证最终用户需要的解决方案能够被成功部署。

最后一个步骤是创建虚拟器件，这个过程包括三个子步骤：第一步，创建虚拟镜像；第二步，分别在虚拟镜像中安装和优化服务解决方案所需的中间件和支撑模块；第三步，安装上文所提到的配置脚本，并且配置相应的脚本执行逻辑和参数，从而使得脚本在虚拟器件的启动、配置过程中能够按照一定的顺行执行。

当与一个应用或服务相关的虚拟器件都创建完成以后，可以将它们保存起来，供发布和部署时使用。

### 3.1.3 发布虚拟器件镜像

随着服务器虚拟化技术的发展，各大厂商都推出了自己的虚拟器件，但是这些产品的接口规范、操作模式互不兼容，妨碍了用户将多个不同厂商的虚拟器件组装成自己所需的虚拟化解决方案，也阻碍了虚拟化技术的进一步发展和推广。在这种背景下，需要统一的标准来明确接口规范，提高互操作性，规范各大厂商的虚拟器件组装和发布过程。

在IBM、VMware、微软、思杰和英特尔等虚拟化厂商的倡导下，DMTF（Distributed Management Task Force）非赢利标准化组织制定了开放虚拟化格式（Open Virtualization Format，OVF）。

OVF标准为虚拟器件的包装和分发提供了开放、安全、可移植、高效和可扩展的描述格式。OVF标准定义了三类关键格式：虚拟器件模板和由虚拟器件组成的解决方案模板的OVF描述文件、虚拟器件的发布格式OVF包（OVF Package），以及虚拟器件的部署配置文件OVF Environment。下面分别介绍OVF描述文件和OVF包，而OVF Environment的内容将在3.2节中介绍。

每个虚拟化解决方案都能够通过一个OVF文件来描述。目前，最新的OVF 1.0规范中定义了虚拟器件的数量，以及每个虚拟器件的硬件参数信息、软件配置参数信息和磁盘信息等各种信息。图3.7描述了一个OVF描述文件的实例结构。OVF描述文件通过对标准的XML格式进行扩展来描述一个虚拟器件（在OVF规范中称为Virtual System），或者若干个虚拟器件整

合成的一个解决方案（在OVF规范中称为Virtual System Collection），这些虚拟器件可以来自不同厂商。由于OVF描述文件中包括了整合后的各个虚拟器件之间的关联关系、配置属性和启动的先后顺序等关键信息，因此用户或者任何第三方厂商编写的部署工具都能够解析OVF文件，并快速地部署其中描述的各个虚拟器件。

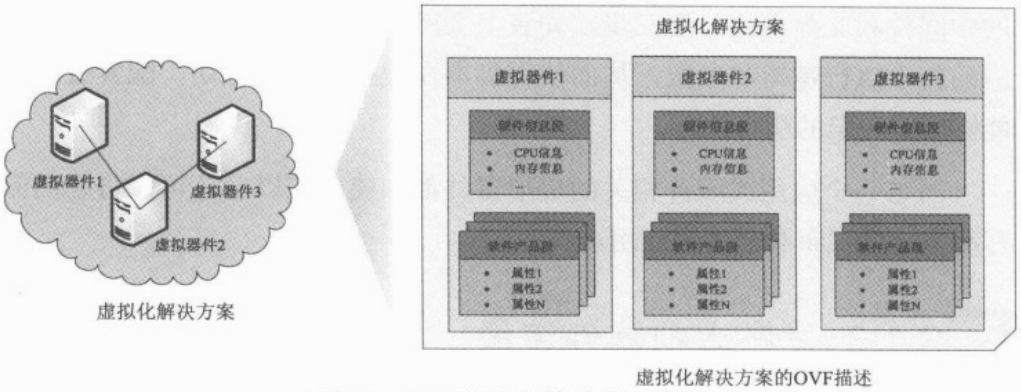



图3.7　OVF描述文件结构示意图

OVF包是虚拟器件最终发布的打包格式，它是一个按照IEEE 1003.1 USTAR POSIX标准归档的以.ova为后缀的文件。OVF包里面包含了以下几种文件：一个以.ovf为后缀结尾的OVF文件、一个以.mf为后缀结尾的摘要清单文件、一个以.cert为后缀结尾的证书文件、若干个其他资源文件和若干个虚拟器件的镜像文件，如图3.8所示。如前所述，OVF文件描述了整个解决方案的组成部分，以及每个组成部分的内在特性和组成部分之间的关联关系。镜像文件既可以是虚拟器件的二进制磁盘文件，也可以是一个磁盘配置文件，它记录了下载二进制磁盘文件的URI地址。摘要清单文件记录了OVF包里面每个文件的哈希摘要值、所采用的摘要算法（比如SHA-1、MD5）等信息。证书文件是对摘要清单文件的签名摘要，用户可以利用这个摘要文件来对整个包进行认证。资源文件是一些与发布的虚拟器件相关的文件，比如ISO文件等。这些文件中，摘要清单文件、证书文件和资源文件是可选的，而OVF文件和镜像文件是必需的。

以OVF包的方式发布虚拟器件，包含以下几个步骤。第一，创建需要发布的虚拟器件所对应的OVF文件。第二，准备好需要添加到OVF包里的虚拟器件镜像，为了减小OVF包的体积，二进制格式的虚拟磁盘可以采用GZIP格式进行压缩。第三，为了防止恶意用户对发布的OVF包进行篡改，应该对OVF包里面的文件做哈希摘要和签名，并将这些信息保存到摘要清单文件和证书文件，但是这个步骤目前并不是必须的。第四，如果有必

要，准备好相关的资源文件。最后，用TAR方式对OVF文件、虚拟器件的镜像文件、摘要清单文件、证书文件和相关资源文件进行打包，并放置在一个公共的可访问的空间，准备被用户下载或部署。

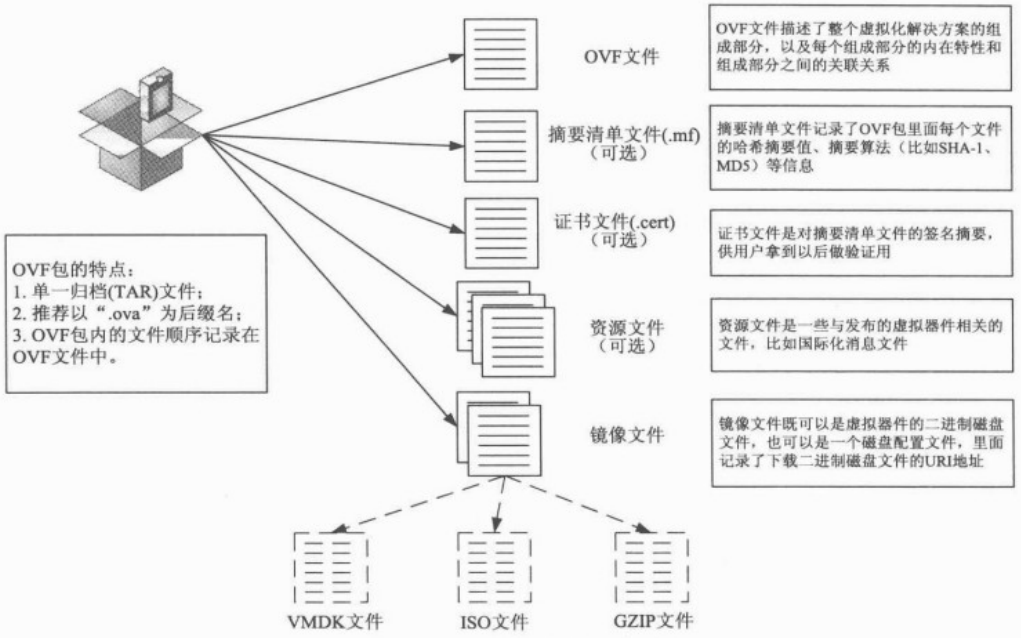


图3.8　OVF包结构示意图

为了简化组装、发布虚拟器件的操作，IBM公司发布了OVF工具箱，它是一个Eclipse插件程序，功能包括可视化地创建、编辑OVF，对OVF所含信息进行完整性校验，以及将虚拟器件打包成OVF包格式。VMware公司也推出了一款叫做VMware Studio的产品，该工具在基于网页的控制台上为虚拟器件创建OVF包，还能够为已经部署的OVF镜像包提供自动更新。思杰（Citrix）公司于2008年底也发布了支持OVF的工具Kensho（预览版），该软件能够将虚拟器件打包成OVF包，并将其导入到多种虚拟化管理平台上。此外，思杰公司还和Amazon公司合作，将该工具应用于Amazon EC2云计算平台上。

### 3.1.4 管理虚拟器件镜像

如上面几节所述，用户按照流程创建、打包好虚拟器件镜像后，会将镜像发布到公共的可访问的仓库，准备被下载或部署。这样的公共仓库会储存大量的虚拟器件镜像，而一般来说一个虚拟器件镜像文件都有几GB甚至几十GB，在这种情况下，对大量虚拟器件镜像的有效管理显得十分重要。

镜像文件管理的目标主要有三个：一是保证镜像文件能够被快速地检索到，二是尽量减小公共仓库的磁盘使用量，三是能够对镜像进行版本控制。目前比较成熟的解决办法是对镜像文件的元数据信息和文件内容分别存储。镜像文件的元数据信息主要包括文件的大小、文件名、创建日期、修改日期、读写权限等，以及指向文件内容的指针链接。而镜像文件的实际内容，一般会采用切片的方式进行存储，将一个很大的镜像文件切成很多的小文件片，再将这些文件片作为一个个的文件单独存放，为每一个文件分配一个唯一的标识符，以及文件内容的摘要串。这需要在镜像文件的元数据里增加新的信息，这个信息记录了镜像文件对应的各个文件片。采用文件切片方法的好处在于，由于很多镜像文件具有相似的部分，例如相同的操作系统目录，通过镜像切片及生成的内容摘要，镜像管理系统可以发现这些镜像文件中相同的文件片，然后对这些文件片进行去重操作，在文件系统中只保存单一的切片备份，这种方法可以大大地减少镜像文件的磁盘空间占用量。文件切片同样有利于镜像的版本管理，因为一般来说，一个文件的版本更新只涉及整个文件的一小部分，通过镜像切片技术，当一个镜像的新版本进入系统时，系统会通过切片及生成摘要，识别出新版本中哪些切片的内容与之前的版本不同，然后只保存这些不同的切片。

在采用了文件切片和版本管理的镜像管理系统上获取一个虚拟器件镜像的流程大致如下：第一步，用户选择虚拟器件的名称或标识符，以及虚拟器件的版本号码，如果用户没有给出版本号码，系统会默认用户需要最新版本；第二步，系统根据用户给出的虚拟器件名称或标识符，在镜像文件库中找到对应的元数据描述文件；第三步，根据用户给出的或由系统生成的版本号码，在元数据文件中找到对应的版本信息；第四步，系统根据元数据文件对应版本中标明的文件切片信息，从文件切片库中找到对应的切片；第五步，系统根据元数据文件中文件切片的顺序，对找到的文件切片进行拼接；第六步，系统将组装好的虚拟器件镜像文件包返回给用户。

### 3.1.5 迁移到虚拟化环境

在虚拟化广泛普及之前，数据中心的绝大多数服务都部署在物理机上。随着时间的推移，这些物理设备逐渐老化，性能逐渐下降，所运行的服务的稳定性和可靠性都受到了极大的影响。然而，想要把服务迁移到新的系统上会面临很大的风险。这主要有两个方面原因：一方面是开发人员

的流动性，当需要迁移服务时，可能已经找不到以前开发团队的相关人员了；另一方面是服务对系统的兼容性问题，服务所依赖的老系统的特定接口或者函数库在新的系统里面并不一定兼容，这些问题长期困扰着传统数据中心的管理。

随着虚拟化的日益流行和其优势的不断体现，人们也在思考如何让已有的服务迁移到虚拟化环境里来充分利用虚拟化所带来的好处。虚拟化的辅助技术P2V（Physical to Virtual）成为了决定服务器虚拟化技术能否顺利推广的关键技术。顾名思义，P2V就是物理到虚拟，它是指将操作系统、应用程序和数据从物理计算机的运行环境迁移到虚拟环境中，如图3.9所示。P2V技术能够把应用服务与操作系统一起从物理服务器上迁移到虚拟环境中，通过这样整体性的解决方案，管理员不再需要触及与系统紧密整合的应用的相关代码，大大提高了系统迁移的可行性和成功率。

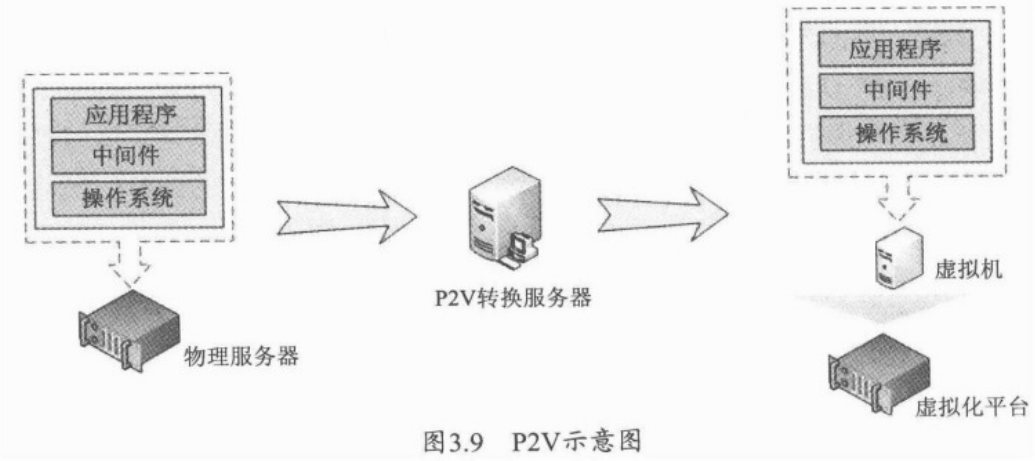



图3.9　P2V示意图

当然，P2V技术的原理并不是文件拷贝那么简单。例如，在操作系统启动过程中，操作系统内核负责发现必要的硬件设备和相应的驱动程序，如果内核没有发现合适的驱动，硬件设备就无法正常运行。因此，要将物理机上的整套系统迁移到虚拟机上，硬件设备从“真实的”变成了“虚拟的”，相应的驱动程序也需要替换成能够驱动“虚拟”硬件的程序。

绝大多数实现P2V技术的软件都遵循了上述原理。下面，我们来看看用户操作P2V软件的基本步骤。

第一步制作镜像，通过镜像制作工具将物理机的系统整体制作成物理机的镜像。这里的镜像制作工具既可以是P2V软件自带的，也可以是第三方的软件。

第二步选择驱动，替换掉镜像中与特定硬件设备相关的驱动程序或者

磁盘驱动器，并且保证镜像中新的驱动程序和其他驱动程序在系统初始化时有序启动，以使镜像能够在虚拟环境中运行。

第三步定制配置，用户手动输入必要的参数，例如虚拟机的CPU、内存、MAC地址等，P2V软件根据数据的参数生成能够让镜像被虚拟机监视器所识别的配置文件。

总之，P2V软件需要捕捉物理系统的所有硬件配置、软件配置、磁盘内容等信息，并对与客户环境定制化相关的配置参数进行抽象，将所有这些信息打包成一个镜像及相应的虚拟机监视器相关的配置文件。

就具体的操作系统而言，由于Linux系统内核是开放的，因此实现P2V的过程相对较为简单；Windows系统内核没有公开，P2V相对比较复杂，如果不能很好地解决驱动替换，在虚拟机启动时很可能出现不能操作的现象，因此存在一定的风险。

值得一提的是，伴随P2V技术的还有V2P（Virtual to Physical）和V2V（Virtual to Virtual）技术。所谓V2P就是将虚拟机向物理机迁移，类似于我们日常所用的Symantec Ghost软件，只是增加了对各种不同物理平台的硬件设备的驱动支持。而V2V技术使得系统和服务可以在不同的虚拟化平台之间进行迁移，比如现有的系统和服务运行在Xen虚拟机上，通过V2V迁移，使得系统和服务可以运行在VMware ESX虚拟机上。



## 3.2 部署虚拟化解决方案



当虚拟器件被创建、发布以后，它们需要通过某种方式被部署到数据中心里才能被用户使用。在这个阶段，我们首先要考虑如何规划虚拟化环境，选择合适的虚拟化厂商和产品，将数据中心的计算资源、存储资源和网络资源进行虚拟化，从而保证虚拟器件能够在虚拟化环境里面正常运行，这些内容将在3.2.1小节“规划部署环境”中讲述。3.2.2小节“部署虚拟器件”将介绍把虚拟器件部署到虚拟化环境里面的具体步骤及相应的关键技术。最后，在3.2.3小节“激活虚拟器件”中将介绍在虚拟器件内部对于虚拟器件模板进行实例化的过程和技术。通过这三个过程，虚拟器件就可以最终被用户使用了。

### 3.2.1 规划部署环境

通过第2章的讲述我们知道，数据中心采用虚拟化能够显著地提高服

务器利用率，缩短服务部署时间，减少能耗、制冷和维护等成本。然而不可否认的是，虚拟化技术同时带来了新的问题：在管理层次上增加了虚拟机层，增加了资源管理和调度的复杂性。另外，面向服务的架构（Service Oriented Architecture，SOA）催生了大量的由松散耦合的功能模块组成的业务，当这些业务被部署在数据中心时需要更加快捷、便利。因此，在数据中心构建虚拟化环境时，用户应该进行投资回报分析，根据自己的业务需求来规划数据中心的计算资源、存储资源和网络资源，并选择适合的虚拟化厂商和产品来寻找虚拟化环境的管理能力及成本的平衡点。

下面将根据构建虚拟化环境的三个步骤即投资回报分析、资源规划和虚拟化平台厂商及产品的选择来分别介绍相关的关键技术。

第一个步骤是投资回报分析。作为企业的管理人员，最关心的是自己的投资能否获得更高的回报，对数据中心实施虚拟化同样要考虑这样的问题，在实施虚拟化之前进行投资回报（Return On Investment，ROI）分析就显得尤为重要。投资回报分析是通过一系列的经济学方法对数据中心内各种资源的成本进行处理分析，得到数据中心实施虚拟化以后效益是否能够提高的预测值。通常，在分析过程中需要考虑直接投资成本和间接投资成本。比较常见的直接投资成本包括：服务器硬件设备成本、网络硬件设备成本、存储设备成本、配套制冷设备成本、虚拟化软件成本、构建虚拟化环境的时间成本和相关设施的维护成本等。另外，还需要结合服务器硬件性能和虚拟化软件来考察数据中心的整体虚拟化能力，这个能力决定了该数据中心能够容纳的虚拟机的数量，从而间接得出能够容纳的虚拟化解决方案数量。很多虚拟化厂商都提供简单的计算工具方便用户计算投资回报率，比如VMware公司的在线ROI计算器、PlateSpin公司的PlateSpin Recon。对于复杂的大型数据中心，用户也可以找第三方的专业公司来分析投资回报率。

第二步是资源规划。数据中心的资源主要包括三大类：计算资源、存储资源和网络资源。计算资源是指物理服务器的计算处理能力，和CPU、内存相关；存储资源是指数据中心的存储能力，和磁带、磁盘、存储系统的空间相关；网络资源是指数据中心的网关、子网、带宽和IP等资源。通过虚拟化技术，数据中心里面的各种资源被整合成了统一的资源池。资源规划就是要研究如何把由虚拟器件组成的解决方案部署在虚拟化环境里，合理分配资源，并且保证资源的高效利用。资源规划一般从计算资源规划入手，资源规划者在能够保证虚拟化解决方案所需要的计算资源的前提

下，再考虑与存储、网络资源池分配相适应的资源。对于计算资源，常用的衡量指标是VM/Core，它指单台物理机的CPU里每个核（Core）上所能运行的虚拟机的数量。如果单台物理服务器的计算资源无法满足解决方案服务的需求，就需要用到多台服务器资源。这时，虚拟机的负载均衡就成为很重要的因素。可以保证规划阶段分配的资源能够得到充分利用。当然，还需要考虑存储资源的I/O负载均衡、网络资源的带宽均衡等。在产品方面，VMware公司推出的资源规划辅助工具Capacity Planner能够帮助数据中心更方便地进行规划。IBM公司的全球技术服务部（GTS）也提供了相关的服务来帮助客户对数据中心现有资产做出评估，并在战略上实施资源规划。

第三步是虚拟化平台厂商及产品的选择。在第二章我们曾简单介绍了x86平台下的主流虚拟化厂商。目前，主流的企业级虚拟化平台有VMware公司的ESX Server、Xen及微软公司的Hyper-V。用户在进行选择时，需要综合考虑这些产品的价格、功能、兼容性，找到适合自己的产品。从价格上来说，VMware ESX Server按服务器的内核数量来计价，Hyper-V是随着Windows Server 2008系统一同发售的，而Xen有两个版本：商业版（Critix XenServer）和开源版，其中开源版可以免费下载和使用。从功能上来说，各个厂商都提供了基本的虚拟化平台及虚拟机管理命令。在这些功能之外，VMware提供了集成化的数据中心管理平台Virtual Infrastructure，以及之上的迁移、容错、备份等套件，XenServer也有对应的数据中心管理工具，微软Hyper-V的附加功能目前比较少。从兼容性上来说，Xen和VMware都对Linux系统有很好的兼容性，在Windows平台下，VMware也能够提供大部分管理功能，并支持创建Windows虚拟机，作为Windows一部分的Hyper-V能够对Windows操作系统提供良好的支持。





### 3.2.2 部署虚拟器件

准备工作完成以后，就可以进行虚拟器件的部署了。部署虚拟器件是将虚拟器件支持的解决方案交付给用户的过程中最重要的一个环节，即虚拟机实例化的阶段。在3.1节所提到的步骤中，我们已经知道了如何创建虚拟器件和发布虚拟器件，而部署阶段所要做的工作就是使虚拟器件适应新的虚拟化环境，并将其承载的解决方案交付给用户。

部署虚拟器件的流程（如图3.10所示）大致可以分为以下6个步骤：

1.选择虚拟器件并定制化；2.保存定制化参数文件为OVF Environment文件；3.选择部署的目标物理机；4.复制虚拟器件的镜像文件和配置文件；5.启动虚拟器件；6.在虚拟器件中进行激活。目前，比较主流的部署工具都能够完成流程中前5步操作，下面我们详细介绍每一个步骤，而第6步操作在虚拟机内部进行，我们将在3.2.3小节中单独介绍。


图3.10　部署虚拟器件流程图

第1步，选择虚拟器件并定制化。在部署虚拟器件之前，用户首先要选择需要部署的虚拟器件，并输入配置参数。这一步是整个部署过程中少数需要用户参与的步骤之一，由于采用了虚拟器件技术，需要用户配置的参数相对于传统的部署已经变得非常简单，而且部署工具还能够帮助用户对这些参数进行配置，进一步减少了用户操作的复杂性。概括来说，用户可以配置的参数信息包括虚拟机的虚拟硬件信息（CPU、内存等），以及少量的软件信息。软件信息是指虚拟机内部软件栈（操作系统、中间件、应用程序）相关的配置，其中与网络和账户相关的参数必不可少。网络参数是连接各个虚拟器件从而构成整体解决方案的重要信息，包括IP地址、子网掩码、DNS服务器、主机名、域名、端口等，它们既可以由用户手动分配，也可以由部署工具自动分配。账户参数的设定是用户定制化最重要的环节，主要包括虚拟机的用户名和密码、某个软件的用户名和密码，或者某个数据源的用户名和密码等。出于安全方面的考虑，这些参数一般情况下需要用户去指定，而不采用默认值。

第2步，保存定制化参数文件。在第1步生成的定制化信息需要保存在文件中，以便被后续的虚拟机配置程序调用。一般来说，定制化信息被保存为两个文件：一个文件保存虚拟机的硬件配置信息，用于被虚拟化平台调用来启动虚拟机；另一个文件保存的是对于虚拟器件内的软件进行定制的信息。虚拟机配置文件与虚拟机的平台相关，因此需要遵循厂商指定的文件格式规范。对于虚拟器件的软件定制化信息，由于在虚拟化技术产生

的初期各个厂商独自开发自己的部署工具，使得保存定制化参数的方式各不相同，例如有些厂商使用文本配置文件，有些厂商使用XML文件。在上文提到的开放虚拟化格式（OVF）成为工业标准以后，这一问题得到了有效的解决，目前各大厂商都会按照OVF Environment文件的格式来保存定制化的信息。在3.1.3小节中已经介绍了OVF标准及其定义的文件格式，具体对于OVF Environment文件，OVF标准是这样定义的：该文件定义了虚拟机中的软件和部署平台的交互方式，允许这些软件获取部署平台相关的信息，比如用户指定的属性值，而这些属性本身是在OVF文件里定义的。OVF Environment规范分为两个部分，一个是协议部分，另外一个是传输部分。协议部分定义了能够被虚拟机上软件获取的XML文档的格式和语义，而传输部分定义了信息是怎样在虚拟机软件和部署平台上通信的。综合来说，虚拟器件的模板描述信息、能够被用户配置的属性项信息、属性的默认值等信息在OVF文件里进行了描述，而客户在第1步填写的定制化信息在OVF Environment文件里面描述。两个文件通过将属性的名称作为关键字进行匹配。

第3步，选择部署的目标物理机服务器。目标机至少需要满足下列几个条件：网络畅通、有足够的磁盘空间放置虚拟镜像文件、物理资源满足虚拟机的硬件资源需求（CPU、内存数量足够）、虚拟化平台与虚拟器件的格式兼容（例如Xen平台支持Xen虚拟器件、VMware平台支持VMware虚拟器件）。目前的部署工具都能够自动完成对上述几个条件的检查工作。具体来说，部署工具会通过网络连接目标服务器，连接成功后，通过执行系统命令检查服务器上的CPU、内存、磁盘空间、虚拟化平台。在检查通过后，返回给用户可以部署的信息。另外，有些部署工具可以提供更高级、更智能的部署能力，让用户事先输入一组服务器的列表，组成一个服务器池，当用户选择要部署一个虚拟器件时，部署系统根据上述几个条件自动从服务器池中选择出满足条件的一台服务器，作为部署的目标机。部署工具还可以考虑用户的定制化需求，将虚拟器件部署到网络较好的服务器，或者部署到硬件性能比较好的服务器，或者部署到没有运行其他虚拟机的服务器，或者考虑一个解决方案中的多个虚拟器件的关系，将它们部署到同一个服务器或者多个不同的服务器上。

第4步，拷贝虚拟器件的相关文件。在用户完成参数定制化并选择了目标物理机以后，部署工具就可以从虚拟器件库中提取出用户选择的虚拟器件的OVF包，再将他们与第2步生成的OVF Environment文件、虚拟机配置

文件一起拷贝到目标物理机上。由于虚拟器件镜像的大小一般都在几GB到几十GB，而目前的网络主要是百兆网或者千兆网，因此部署的时间瓶颈在于传输所耗费的时间。随着虚拟化服务越来越受到人们的重视，相应的厂商也不断开发出新的技术来解决部署费时的问题，目前比较成熟的技术有镜像流技术和快照技术。

镜像流传输类似于在线视频播放的流媒体：通过流媒体技术，用户可以边下载影音文件，边播放已下载的部分。这样的好处是用户不需要等待整个文件下载完毕再播放，节省了时间，优化了用户体验。对于典型的虚拟器件，其内容包括操作系统、中间件、应用软件，以及用户需要使用的剩余空间。用户在启动虚拟器件时，主要是启动虚拟器件的操作系统、中间件和应用软件，这些部分仅占整个虚拟器件文件中的一小部分，通过镜像流技术就可以无需下载整个虚拟器件而即时启动虚拟机。简单来说，在虚拟器件启动时，虚拟器件通过流传输的方式从镜像存储服务器传输到虚拟化平台上，虚拟器件在接收其镜像的一部分后，即可开始启动过程。虚拟器件余下的部分可以按需从镜像存储服务器中获取，从而减少了虚拟器件的部署时间，使得部署的总时间只需要几十秒钟到几分钟。镜像流传输技术与3.1.4小节中提到的镜像切片技术可以很好地结合，部署系统按照流传输方式请求镜像时，镜像管理系统无需将文件片打包成镜像文件包再整体返回给部署工具，而是按照文件片的顺序，依次将文件片以文件流的方式传输给部署工具。通过省去虚拟器件文件片组装打包的过程，进一步缩短了整个部署的时间。

快照技术的本意是用来帮助虚拟机进行备份和恢复，但是它同样可以辅助虚拟化服务的部署。快照技术在部署中的典型应用场景是：在部署虚拟器件时，部署工具会检查在部署目标机上是否已经存在被部署虚拟器件的快照，如果存在，就不需要再将虚拟器件镜像文件拷贝到虚拟化平台，而是通过虚拟化平台的应用接口将快照作为模板，快速复制出新的虚拟器件，并通过定制化配置成为用户可用的状态；如果快照不存在，在虚拟器件镜像被部署后，部署工具会通过虚拟化平台提供的应用接口对虚拟器件做快照，方便以后使用。快照技术的好处在于可以减少二次以至多次部署的时间。

第5步，在目标机上启动部署后的虚拟器件。部署工具会通过远程连接的方式，在目标机上执行一组命令，来完成虚拟器件的启动。在启动过程中有一个关键过程，是将第2步生成的软件配置参数文件传送到虚拟器件

中。目前采用虚拟磁盘的方法进行传送，也就是说将OVF Environment文件打包为一个ISO镜像文件，在虚拟器件的配置文件中添加一个虚拟磁盘的配置项，将其指向打包的ISO镜像文件。这样，当虚拟器件启动后，在虚拟器件内部就可以看到一个磁盘设备，其中存放着OVF Environment文件。总体来说，这一步需要执行的操作依次为：将OVF Environment文件打包为ISO文件，修改虚拟器件配置文件创建虚拟磁盘项，在虚拟机管理平台上注册虚拟器件信息，启动虚拟器件。

### 3.2.3 激活虚拟器件

虚拟器件部署的最后一个步骤是在虚拟器件内部读取OVF Environment文件的信息，根据这些信息对虚拟器件内的软件进行定制，这个过程被称为虚拟器件的激活（Activation）。根据激活的自动化程度及功能，激活可以划分为：完全手动的激活、基于脚本的手动激活、单个虚拟器件的自动激活、组成解决方案的多个虚拟器件的协同激活。下面将分别介绍这几种场景。

完全手动的激活适用于所有的虚拟器件，用户在虚拟器件内部读取OVF Environment文件的内容，判断其中的配置项属于哪个软件，并根据自己的知识对该软件进行配置。显然，这种场景对用户的要求较高，要求用户了解OVF Environment文件的格式，能够读懂其中的内容，并具备对各种操作系统、中间件、应用软件进行配置的知识，即使用户具备这些知识，但是由于配置过程非常复杂，也可能因为误操作或者系统异常终止而导致激活失败。

3.1.2小节中介绍的脚本技术可以简化激活的过程。脚本是由虚拟器件的创建者、发布者编制的，在激活过程中，用户只需要调用配置脚本，并将OVF Environment文件中的配置信息作为脚本的输入参数，就可以完成激活，用户不需要了解激活脚本的工作流程，因此也不需要具备对各种软件产品进行配置的知识。不过这种方式对用户仍有一定的要求，一是用户需要读懂OVF Environment文件的内容；二是用户需要了解激活脚本暴露的接口格式，并将OVF Environment文件对应的内容传给脚本；三是用户需要了解并协调多个脚本的执行过程，因为在激活中，多个软件的激活可能需要遵循一定的顺序。而下文介绍的自动化激活问题，正是为了满足上面的几个要求。

一个典型的自动化激活单个虚拟器件的工具的工作原理如下：在虚拟器件启动过程中，激活工具从虚拟磁盘中获取OVF Environment文件，根据激活的先后顺序读取OVF Environment文件中的参数，执行激活脚本，配置虚拟器件中的软件，在不需要用户干预的情况下，得到定制化的可用的虚拟器件。这样的部署方式改进了传统的软件安装和部署方式，免去了那些费时并且容易出错的部署步骤，比如编译、兼容性和优化配置，并且这种方式在虚拟资源池智能管理的支持下能够做到完全自动化，非常适合在虚拟化环境中对软件和服务进行快速部署。目前，很多公司开发的虚拟器件都内置了简单的激活工具，例如IBM Activation Engine作为一个自动化激活工具，在IBM公司发布的虚拟器件中得到了广泛使用。

在3.1.2小节中我们提到，多个虚拟器件会组合成一个解决方案，而在激活过程中，这些虚拟器件可能有配置参数的依赖关系和激活顺序关系。通过在虚拟器件内部植入具备网络通信功能的激活工具，可以统筹整个解决方案的激活过程，协作地完成解决方案的激活。当然，这需要借助现有的OVF文件中定义的参数依赖关系及激活顺序。

## 3.3 管理虚拟化解决方案

数据中心的管理需要资源的自动化调度和与业务相关的智能。一个数据中心好比一个交响乐队，每一个业务和它所占有的资源就好比一个乐手和他的乐器，乐手必须熟练运用好乐器才能演奏出美妙动人的独奏。乐队里面有弦乐、管乐和打击乐三大声部，包括数十乃至上百件乐器，如果不能很好地协调在一起，即使每个乐手都是世界一流的，整个乐队演奏出来的也是毫无组织，杂乱无章的。因此，乐队需要一个指挥家，作为整个乐队的灵魂，将乐队的各个部分组织起来，对各个声部进行有序地调度，形成一个整体呈现给听众。同样，现代数据中心既需要单个业务能够自治管理，也需要一个负责全局控制和协调的中心节点（Orchestrator）对数据中心的业务和资源进行统一监控、管理和调度。在传统的服务管理模式中，管理员需要登录若干个软件的控制台来获取信息、执行操作，这种分别针对软件、硬件和系统的方式缺乏面向服务的统一视图。而采用虚拟器件后，管理员可以通过虚拟化平台提供的管理功能来完成对虚拟机的管理工作，例如开关虚拟机、调整虚拟机资源、执行实时迁移等，也可以通过虚拟器件内部嵌入的管理模块来管理解决方案，如服务监控、服务开停控制、服务自动性能调优等。这两类管理操作都可以被统一到集中式的管理平台中。

在虚拟化环境里面，不仅仅需要实时监测宿主机的电源和性能的变化，还需要了解虚拟机CPU和内存的利用率，甚至是业务的访问量，这些信息对于资源管理和调度是至关重要的。采集到这些信息以后，中心节点会根据应用特征选择最合适的调度算法，将这些信息抽象成该算法的输入，计算出最优化的调度结果，之后按照调度结果对虚拟机进行调度。除此之外，数据中心管理程序还需要考虑各种常规的管理操作，例如开关、配置等，通过对流程的自动化来简化数据中心管理员的工作。

本节将介绍虚拟器件管理阶段的四类关键技术：集中监控、快捷管理、动态优化和高效备份。

### 3.3.1 集中监控

虚拟化技术为数据中心带来先进的功能已是不争的事实。但是，由于引入了虚拟化，对数据中心资源的管理和监控任务也随之增多。传统的数据中心大致分为硬件、操作系统、中间件和应用四层。引入虚拟化以后，一台物理服务器上会运行多个虚拟机，这使得硬件和操作系统之间又多了一个层次，数据中心需要管理维护的对象的数量和复杂度也增加了。数据中心的管理平台中需要能够对虚拟化环境进行集中监控的技术，以便更好地监控虚拟化环境中的资源及运行在虚拟器件上的解决方案。数据中心的管理平台在监控方面必须做到以下两点：第一，能够集中监控数据中心的所有资源；第二，能够集中监控所有虚拟器件上运行的解决方案的状态和流程。下面我们分别阐述这两点所涉及的技术。

对所有资源集中监控，就是通过对采集到的数据进行分析、优化和分组，以图表等形式，让管理员在单一界面对虚拟化环境中的计算资源、存储资源和网络资源的总量、使用情况、性能和健康状况等信息有明确、量化的了解。比如，对于每个物理服务器，管理员要能看到它的CPU和内存的使用情况、它上面运行的虚拟机数量，以及每个虚拟机的负载情况、所占用的IP资源、带宽资源等。其次，管理员还要能够监控各个物理机上的虚拟机的拓扑结构图，以及虚拟机和物理服务器的位置关系等。另外，通过资源集中监控，还能帮助管理员发现负载不均衡的情况，以及排除故障。在集中监控方面，IBM的Systems Director通过Virtualization Manager扩展功能（如图3.11所示），可以让管理员快速地查看数据中心的物理机、虚拟机、存储设备等资源的数量、健康状况、逻辑关系等，另外还可以让

管理员定制视图，从而进一步获得更详细的信息。

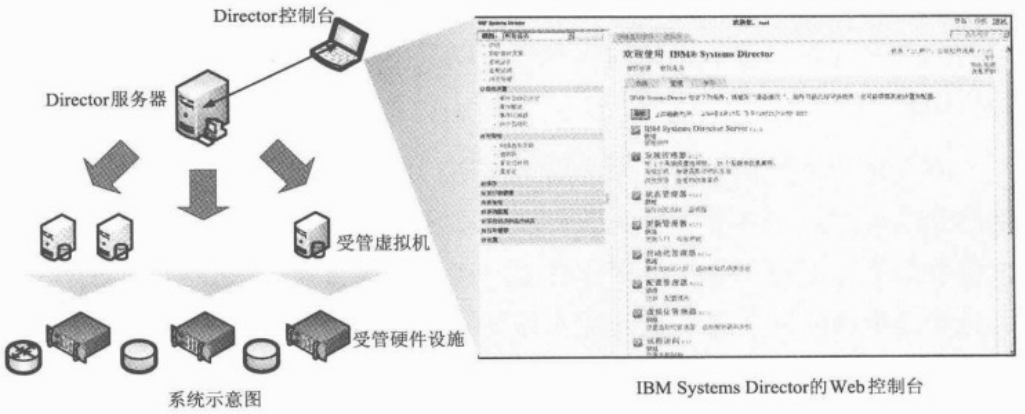



图3.11　IBM Systems Director集中控制台

对虚拟器件上运行的解决方案的状态及流程进行集中监控，首先要能够让用户实时跟踪这些解决方案在部署及运行期间的状态和流程的实时情况。虚拟化服务从被部署开始，要经历多个状态，包括部署、激活、管理，直到最后生命周期结束而被销毁。虽然部署与激活的流程可以根据用户的配置自动完成，但是仍要求有一个集中的可视化监控环境为用户提供他们所关心的信息。如部署所采用的虚拟器件包、预留的物理资源、部署（虚拟器件文件传输）的进度等。在激活过程中，这些信息包括解决方案的配置及激活操作的结果等。同样，当解决方案经过激活运行起来以后，管理员所关心的主要有解决方案的性能信息，包括它所提供的服务的响应时间、吞吐量等，以及每个虚拟器件的运行状态，如虚拟CPU、处理器和磁盘的使用率等。将这些信息以可视化的方式展现给管理员，他们便可以有的放矢地对在数据中心中托管的虚拟器件及解决方案进行管理和调优。最后，当虚拟器件完成了其任务并准备被销毁时，其销毁的过程及销毁后的状态也需要进行监控，来帮助管理员完成对虚拟器件整个生命周期的管理，并保证所有的资源被有效地回收。

## 3.3.2 快捷管理

在数据中心中，管理员或者管理程序下达的管理指令主要针对三种类型的实体：第一类是基础设施，常规操作有开关物理机、配置网络等；第二类是虚拟机，常规操作有开关虚拟机、调整虚拟机资源、迁移虚拟机、进行快照操作等；第三类是虚拟器件内的应用、软件、解决方案，常规操

作有开关软件、配置软件等。如何将这三类实体涉及的多种管理操作简化、流程化、自动化，就是简化管理要解决的问题。具体来说，简化管理可以分为物理机和虚拟机的简化管理，以及虚拟器件内部应用的简化管理两个问题。

对于虚拟器件内部的应用和解决方案，简化管理需要借助虚拟器件内部的管理模块，这些管理模块既可以在创建虚拟器件时安装，也可以在虚拟器件部署以后由部署工具植入进去。这些模块与数据中心管理程序中的简化管理模块协同工作，来完成大部分简化管理的操作。举例来说，一个典型的管理操作是启动一个虚拟化解决方案，如果对于3.1.2小节中描述的Trade应用，传统的数据中心管理员需要顺序执行以下操作：分别启动三个虚拟器件所在的物理机，分别启动三个虚拟器件，启动DB2应用，启动WAS应用，启动IHS应用。这样需要9步操作，同样，对于一个应用的关闭也要按相反的顺序执行9步操作。在采用了简化管理技术之后，这个启动或关闭流程都通过元数据的方式描述并存储在管理程序中，管理程序会解析元数据中的信息，并自动按序执行开关命令，管理员只需要发出开启或关闭应用的一个指令。

对于物理机、虚拟机的简化管理，需要考虑的主要是能够与各种物理机、虚拟机平台进行通信，发出指令。对于物理机的简化操作可以使用很多的现有技术，因而虚拟机简化操作成为了这部分的重点研究方向。目前，为了支持在虚拟化平台上进行二次开发及第三方的管理，主流的虚拟化供应商都适时推出了软件开发包（SDK）和开放编程接口（API）以满足用户自身定制的需要。VMware和Xen目前都有比较成熟的开放编程接口，并被业界其他厂商广泛调用。但是，如果虚拟环境里面有多种虚拟化平台，每个虚拟化平台都有自己的软件开发包，就会对统一管理带来很大的麻烦。因此，对于虚拟化平台的操作也应该标准化，比如开关虚拟机、查看虚拟机状态和资源使用情况、调整虚拟机资源、对虚拟机进行实时迁移等，如果这些操作在不同的虚拟化产品上有不同的实现和格式，用户使用将有很大的不便。目前，业界正在开发一套支持多种虚拟化平台的通用API集合，功能包括：得到虚拟机平台所在物理机的资源状况、开关虚拟机、监控虚拟机状态和资源使用情况、调整虚拟机资源、进行虚拟机实时迁移等。用户通过访问这组API，可以进行与虚拟机、虚拟化平台相关的大多数操作，而无需关心下层虚拟化平台的特殊性。

### 3.3.3 动态优化

自虚拟化诞生以来，采用虚拟化技术的服务性能便一直是用户和相关研究人员关心的重要问题。因为相对物理机，虚拟机的性能有少量的下降，尤其是I/O密集型应用的性能下降会稍大一些。不过，经过大量的实践测试，只要在虚拟化环境中采用动态优化技术，并且配合虚拟化带来的灵活性、资源抽象等优势，不仅可以完全弥补采用虚拟化可能带来的性能下降，而且能为客户带来更多的好处。

动态资源优化技术研究的问题是：在虚拟化环境中，如何根据应用、服务负载的变化为其所在的虚拟机及时、有效地分配虚拟化环境中的资源，保证既不会因为资源缺乏而影响业务系统运行，也不会造成严重的资源浪费。为了使虚拟机的资源达到供求平衡，动态资源优化技术需要了解和掌握各个应用、服务可能的负载量，根据一定的方法或规则推算出其需要的物理资源类型及数量；在应用、服务运行中实时监测其性能数据，预测业务变化的趋势，做出资源再分配的决策，然后进行相应的调整。

动态优化技术需要两只“眼睛”、一个“大脑”和两只“手”来协同工作。通过先看后想再动手的方式完成每一个优化周期，通过定期优化来获得用户期望的性能和资源供求的动态平衡。具体来说，一只“眼睛”从虚拟化平台的角度进行资源监测，了解虚拟环境下有多少台服务器及他们的资源状态，包括CPU、内存、存储和网络等资源的总数量和剩余数量；另一只“眼睛”从应用、服务的角度进行监测，了解在当前虚拟化环境中运行的所有应用、服务的负载状况，以及相应的资源使用情况。这两只“眼睛”分别从供给面和需求面对资源进行监测。一只“手”做宏观调整，即通过打开或者关闭服务器，或利用实时迁移技术移动虚拟机等，调整虚拟化环境中服务器的计算资源；另一只“手”做微观调整，负责调整某个服务、应用所在的部分或全部虚拟机的计算资源，比如调整虚拟机的CPU数量和内存使用量等。所谓一个“大脑”就是具备性能分析预测、进行资源动态规划和输出调度结果的算法，它协调着两只“眼睛”和两只“手”。在优化过程中，首先，它通过两只“眼睛”得到虚拟化平台的计算资源使用情况、应用负载情况；然后根据当前情况并结合历史信息预测应用未来的负载状况，根据预先定义的规则做出资源分配的决策，并进而输出资源调度指令；最后，通过两只“手”来完成调度，资源分配变化不剧烈的时候只需要第二只“手”做微观调整即可，而变化剧烈时需要用上

第一只“手”。“大脑”是整个动态优化技术的核心，大脑的智能程度决定了虚拟环境是否能有效地保证每时每刻都能向应用、服务提供充足的计算资源。

动态优化的“大脑”可以采取多种成熟的调度算法，但本质上讲他们都是一种在决策空间中的搜索算法。搜索算法可采用贪心算法、分而治之算法或启发式算法等。需要指出的是，“大脑”需要考虑的主要因素包括：了解虚拟化解决方案负载的变化规律，实现预动性的调整；了解虚拟化解决方案的服务级别协定（Service Level Agreement，SLA），尽可能满足SLA的需求；合理设定资源池，使得虚拟机只在限定的范围内移动，简化优化复杂度；尽可能将资源消耗互补的虚拟机放在同一台物理机上，使得物理机的资源能够得到更充分的利用；尽可能将构成集群系统的若干台虚拟机放在不同的物理机上，使得物理机发生故障时集群系统不会完全瘫痪；准备适量的后备资源，当出现突发事件时可以立即启用后备资源，保证服务的正常运作。

目前，实现了动态优化技术的产品有：VMware的分布式资源调度器（DRS，Distributed Resource Scheduler），它已经集成在了VMware虚拟化产品Virtual Infrastructure 3内；PlateSpin的PowerRecon，它采用了新的组合编制模块，使得数据存储能够尽量的小；微软也有望在其虚拟管理工具Virtual Machine Manager中推出类似的技术。

数据中心的动态资源调度在执行时所使用的核心技术是在第2章中介绍过的实时迁移技术，它使得数据中心运行的业务具有很强的可伸缩性。由于数据中心的服务器性能不尽相同，并且虚拟机所承载的业务在不同时间访问量也有变化，因此利用实时迁移技术可以实现根据业务流量的变化实时调整虚拟机所占用的资源。下面将介绍目前各个虚拟化厂商比较成熟的实时迁移技术。

在Xen平台下，虚拟机的启动方式可以是本地启动，也可以是网络启动。而Xen上面的实时迁移技术的前提是虚拟机是通过网络方式启动的，即镜像被放置在共享存储上，而虚拟机是运行在宿主机上的（使用该宿主机的CPU和内存），宿主机上的虚拟机管理程序利用NFS或者其他网络存储方式获得镜像上的数据。经过简单的配置，虚拟机就可以实时迁移到另外一台宿主机上。迁移的过程对用户来讲是透明的，迁移后虚拟机使用新宿主机的计算资源，而文件系统所在的镜像仍然在共享存储上。

VMware VMotion是VMware公司开发的实时迁移功能，该功能包含在VMware企业级产品VMware Infrastructure中。VMotion和Xen的实时迁移技术原理上十分相似。不同之处在于VMware公司开发了一个文件系统VMFS，这种集群式文件系统允许多个VMware ESX服务器同时访问虚拟机的镜像，因此在迁移过程中无需移动镜像，只需要将虚拟机内存中的状态和上下文通过高速网络从源ESX宿主机传输到目标ESX宿主机。

小型机和大型机环境下的虚拟化同样提供了功能强大的实时迁移功能。在IBM POWER6架构下的System P570服务器基础上，IBM公司推出了实时分区迁移（Live Partition Mobility，LPM）技术和共享的专用容量技术。实时分区迁移可以将用户正在使用和运行的分区从一台POWER6服务器迁移到另一台POWER6服务器，期间无需停止任何应用程序，这样能够帮助客户避免因计划中的系统维护和工作负载管理而造成应用程序中断。

### 3.3.4 高效备份

在传统的数据中心中，数据备份技术已经相当成熟。如果需要对数据进行短期备份，可以利用磁盘；如果是长期备份，则需要用到磁带库。现有的备份机制和相关软件已经发展到可以支持存储区域网络、光纤和系统升级的功能。各个厂商也都推出了自己的存储管理解决方案，并各具优点。

在越来越多的企业开始采用虚拟化技术的情况下，如何对虚拟化数据中心的数据进行备份成为了一个重大挑战。由于以下几个原因，传统的数据备份技术已经不能满足虚拟化平台下的需求。第一，大量具备高度相似的内容的虚拟机镜像并存。在传统的情况下，文件系统和服务器之间的关系是一对一的，但是引入虚拟化以后，一台服务器上面可能运行多个虚拟机，而每个虚拟机都有独立的文件系统作为支撑。第二，有些虚拟化平台为了构建存储集群，采用了私有的文件系统格式，比如VMware ESX独有的文件系统VMFS。这要求数据备份软件能够识别私有的文件系统，并且有访问权限，这就增加了数据备份的复杂性。第三，如果企业的数据中心采用了多种虚拟化平台，那么数据备份时还需要处理虚拟平台的异构性和多样性。第四，多个虚拟器件才能承载一个解决方案。在企业的数据中心里面，由单一服务器承载单一解决方案的情况越来越少，人们看到更多的是由多个虚拟器件组成一个解决方案交付给终端用户。这样，解决方案和虚拟器件的对应关系是一对多的，而多个虚拟器件可能分布在多个虚拟化

平台上。在这种情况下，传统的备份策略和方法很难奏效。第五，虚拟机可以实时迁移，从文件系统的数据备份角度来讲，很难跟踪到底虚拟机运行在哪台物理服务器上。这些挑战都对数据备份一致性提出了更高的要求。

针对虚拟化对数据备份提出的挑战，人们对备份策略和技术做出了相应调整，主流的备份机制有如下两种。

第一种是虚拟机上备份。这种方法沿袭了传统的备份方法，认为虚拟机是一个普通的服务器，只需要在它上面安装和物理服务器上一样的备份代理软件，与传统的备份服务器通信，并执行由备份服务器发出的备份策略和指令。这个解决方案的优点在于它的实施过程和传统的物理服务器备份一样，最大限度地兼容了传统的备份机制，减少了为升级备份而投入的初期成本。很多企业出于这方面的考虑，也乐于采用这种备份方案。其缺点在于备份冗余度过高，增加了后期存储备份数据设备的开销。造成这种情况的原因是，在虚拟机管理器上进行的备份中，它上面虚拟机的文件系统作为普通二进制文件做了一次备份，而虚拟机作为普通服务器，又对自己的文件系统做了一次备份。时间一长，后期存储所需的开销将会增加，而且由于进行了重复备份，备份时间也相对较长。不过，有些数据备份厂商已经意识到了这个问题，具有识别并删除重复数据功能的备份软件已经问世，它能够大量地减少备份量，从而节省备份时间。

第二种是虚拟机外备份。与第一种方案不同，这个方案是在虚拟机外部实现对虚拟机的数据备份，它充分利用了虚拟机管理器提供的备份应用接口，从而简化数据备份和数据恢复的工作，并且减少了备份过程中对其他虚拟机的影响，大大提高了备份效率。其实，这里提到的备份应用接口就是指虚拟机快照技术，虚拟机快照技术不仅能够针对虚拟机文件系统进行快速备份，而且还能将备份粒度降低到文件系统中的某个具体文件。有了虚拟机管理器提供的备份接口，虚拟机外备份方案只需要关心上层的备份策略，而不用和虚拟化平台特定的文件系统打交道。它的主要备份策略是设置虚拟机的还原点，通过逻辑单元号（Logic Unit Number，LUN）或者磁盘驱动器中指定的位置来存储所需的备份。另外，系统管理员还可以通过快速查询逻辑单元号对应的虚拟机，提高恢复虚拟机的响应速度。对于删除重复数据这一项功能，虽然在虚拟机上备份解决方案中不常应用，但是在虚拟机外备份解决方案中却属于常见功能。备份软件能够先将多次出现的相同数据识别出来，并将冗余数据删除，仅存档一份数据。

在实际的生产环境中，很多数据中心所使用的备份解决方案并没有我们上面阐述得那么明确，而是在这两种解决方案中各取所长，可按照用户的实际需要选择恰当的技术。例如，持续数据保护技术就是利用了前面提到的两种解决方案，采用了增量备份的策略对虚拟化数据中心进行持续、增量的数据备份，从而缩短备份所需时间，并减少存储所需的空间。具体来说，在初始化的时候，该系统对数据中心所有的物理服务器和虚拟机服务器进行一次扫描，然后进行一次初始化备份，这一次备份的时间较长。之后，数据保护系统按照备份策略对服务器进行再次扫描，如果发现服务器的文件发生了变化，该系统会对它进行增量备份，并且记录好时间戳。这样，一旦出现任何问题，持续数据保护系统都可以将状态平滑地回滚到出问题以前某一个进行过数据备份的时刻。增量备份只需要很少的系统开销，几乎不会影响到服务器上运行的应用和服务。

## 3.4 小结

本章以虚拟器件生命周期的三个阶段为主线，介绍了服务器虚拟化的关键技术。在虚拟器件创建阶段，讨论了构成服务器虚拟化的虚拟镜像和虚拟器件的概念及创建、组装、发布的流程和关键技术。另外，还阐述了这方面的业界标准，对虚拟器件镜像文件进行管理的关键技术，以及将运行在物理服务器上的传统服务迁移到虚拟机中的P2V技术。在虚拟器件部署阶段，首先探讨了在数据中心构建虚拟化环境需要考虑的问题和相关的虚拟化产品，然后描述了部署虚拟器件的主要流程、提高部署效率的关键技术，以及激活的关键技术。在虚拟器件的运行期间，为了方便管理员集中管理，以及提高应用的性能，从集中监控、快捷管理、动态优化、高效备份四个方面讨论了相应的关键技术。

CHAPTER 4

# 第4章 虚拟化的业界动态

在信息技术领域，虚拟化无疑是目前最受关注的热点技术之一，特别是在各大公司IT开销不断提高的形势下，虚拟化因其具有能够节约成本的优势而更受重视。根据市场研究机构IDC的预测，虚拟化市场在未来两三年将迎来飞速增长。当前，掌握虚拟化核心技术的IT厂商有哪些？他们各自的发展和技术特点是什么？有哪些最新的业界动态？本章将为读者解答这些问题。

## 4.1 IBM

### 4.1.1 概述

IBM公司从事虚拟化的历史已经超过40年，在虚拟化领域的影响力可以用“广泛”和“全面”来概括，这是其他任何厂商都不具备的优势。早在20世纪50年代末到60年代初，虚拟化的概念就已经开始在计算机学术界萌芽。在1959年6月召开的国际信息处理大会上，Christopher Strachey发表了一篇名为“Time Sharing in Large Fast Computers”的论文。这篇论文被认为是有关虚拟化技术最早的学术论著。世界上第一台采用了虚拟化技术的计算机是20世纪60年代中期由IBM公司在Thomas J. Watson实验室设计和实现的IBM 7044（M44），如图4.1所示。采用虚拟化技术后，在一个IBM 7044（M44）计算机上能够用硬件和软件模拟出多个IBM 7044（M44X）虚拟机。这虽然只是一个研究项目，但是由于它使用了分页、虚拟机和计算性能测试等先进技术，对此后虚拟化技术甚至整个计算机工业的发展都产生了意义深远的影响。20世纪60年代，在著名的System 360系统里，IBM

公司第一次将虚拟化平台（Hypervisor）作为一个商业套件发售。

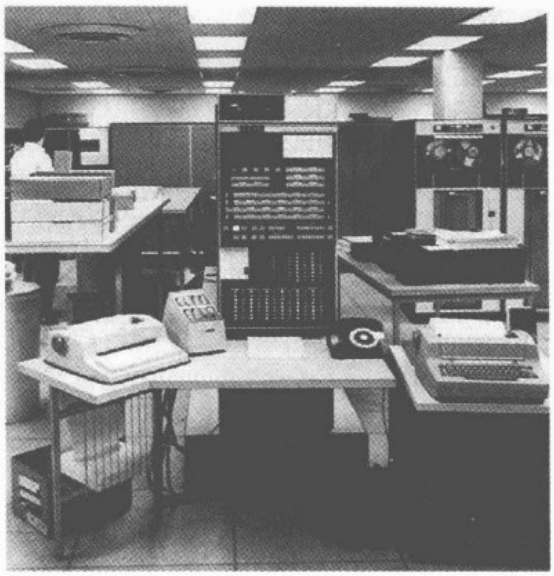


图4.1　IBM 7044（M44）

从产品的角度来讲，IBM公司从大型服务器的虚拟化到应用虚拟化都有相应的产品，能够向用户提供业界最广泛的虚拟化能力。目前IBM的硬件产品线从大型机到微型机分为z、p、x三个系列，在System z和System p系列中都有独立的虚拟化产品，IBM公司所生产的x86服务器、存储设备和网络设备等多种产品也都支持虚拟化。

从虚拟化技术层次来讲，IBM公司在精简指令集计算机（RISC）体系架构上实现了从芯片级、系统级到应用级的全方位虚拟化技术。

从虚拟化解决方案来讲，通过采用IBM跨平台的虚拟化、自动化和系统管理解决方案，用户能够简单、动态地访问和管理资源，可以获得更高的资源使用率和更低的运行成本，从而满足简化基础设施、快速部署应用和提高业务弹性等多种需求。

从虚拟化战略来讲，IBM公司在虚拟化方面具有雄厚的研发实力、丰富的产品线和解决方案，能够给客户提供最广泛、全方位、智能的虚拟化基础架构和解决方案。IBM公司的战略是“虚拟一切资源”。基于这一战略，IBM公司将整合现有的虚拟化技术，在更高层次上实现计算资源和存储资源的虚拟化，在虚拟化的数据中心内实现跨虚拟化平台的、智能的、动态的资源调度，以满足企业客户对IT基础架构动态高效、节能环保和简化管理的新需求，提高数据中心的使用率并降低运行维修成本。可以预见，新一代虚拟化战略带给企业客户的不仅仅是信息技术资源运行模式的

转变，更是一场意义深远的计算革命。

IBM公司的新一代虚拟化战略涉及以下三点。第一，实现数据中心IT资源的虚拟化。通过实现从底层硬件和系统的虚拟化到存储和网络的虚拟化，使数据中心的IT资源成为一个虚拟的资源池，可以按照一定的粒度实现资源的分配，这是实现新一代虚拟化战略的最基础一环。第二，管理和整合虚拟资源。只做到了第一点还达不到新一代虚拟化预期的目标，只有拥有了对虚拟资源的管理能力，数据中心才可以实现自动部署、集中监控、简化管理、动态优化、数据备份等功能。相反，如果没有这种管理能力，所有虚拟化后的资源都只能是一盘散沙。第三，实现元数据和操作流程的标准化。如果虚拟化厂商和解决方案提供者都只按照自己的规范来构建虚拟化数据中心，而不采用统一和开放的标准，那么整个虚拟化产业就很难得到更进一步发展。所以，业界必须大力支持虚拟化的标准化工作，使得整个行业能够在统一和开放的标准下定义各种元数据，并对相关操作流程进行规范。

接下来，我们将从IBM的z系列服务器虚拟化、p系列服务器虚拟化和虚拟化管理三个方面进一步介绍IBM的虚拟化技术。





### 4.1.2 z系列服务器

IBM的System z系列服务器常常被人们称为大型机，它是世界上最成熟的商用服务器，以其无可比拟的高性能、高可用性、高可靠性和高安全性等优点长期服务于银行、电信、公共事业等重要市场。对于一台性能强大的计算机来说，很难有一个应用可以独占所有的资源，因此z系列服务器在设计之初考虑的一个准则就是共享，而虚拟化正是支撑共享的核心技术。z系列服务器的共享涉及计算机系统从下向上的各个环节，包括硬件、虚拟化平台、操作系统、中间件、应用软件、数据等。

1964年，IBM公司Thomas J. Watson实验室的研究人员L. W. Comeau和R. J. Creasy发明了一款名为CP-40的新型操作系统，这是第一款实现了全虚拟化的操作系统。该操作系统采用虚拟内存和虚拟机技术，能够在一台大型机上同时运行多达14个S/360家族的操作系统，从而使得当时还非常昂贵的计算资源能够被充分、有效地利用。

1965年是虚拟化技术发展史上又一个重要里程碑。在这一年，IBM公

司推出了S/360-67系统，它是z系列服务器的鼻祖。该系统实现了虚拟机监视器（Virtual Machine Monitor，VMM），并对所有的硬件接口实现了虚拟化。为了高效利用底层的硬件设备，它可以分时执行多个虚拟机，每一个虚拟机都运行着一个与其他虚拟机相隔离的操作系统，这个操作系统称为VM/CMS（Conversational Monitor System，CMS）。VM/CMS在设计时就考虑到了向后兼容，目前主流的System z9主机上仍然能够使用它。

z系列服务器上的虚拟化既有硬件技术又有软件技术，他们都能够与z系统的体系结构有机、无缝地结合。下面将分别介绍z系统上主要的虚拟化技术，这些技术是目前最成熟而且最稳定的虚拟化技术。

首先需要了解的技术是虚拟机监视器程序PR/SM（Processor Resource/Systems Manager）及其上的逻辑分区概念（Logic Partition，LPAR）。PR/SM整合在z系统中，它将系统的物理资源映射为逻辑资源，供逻辑分区使用。逻辑分区能够将物理机的计算资源划分成若干份，每一个逻辑分区内都可以运行虚拟机，PR/SM使这些虚拟机彼此之间保持独立。PM/SM可以将逻辑分区的资源占有方式配置为独享式或共享式。独享式逻辑分区会独占处理器资源，即使它目前没有全部占用甚至完全没有使用这个处理器，该处理器也不会被其他逻辑分区使用，这样的好处是PR/SM的资源调度压力比较小；而共享式逻辑分区运行在没有被独享逻辑分区占有的处理器上，可以更灵活地使用资源，当然也给PR/SM带来了额外负载。

除了逻辑分区，z系统又引入了地址空间隔离的技术来保证访问逻辑分区的内存不会出现冲突。系统给每一个逻辑分区分配一段单独的地址空间，当程序发生问题时，错误会被隔离在逻辑分区自己的地址空间内，而不会影响其他地址空间。这大大提高了系统的可靠性，减少了错误恢复时间。

在网络支持方面，z系统提供了HiperSocker技术。它保证在z服务器上运行的z/OS、z/VM、Linux、z/VSE等虚拟机系统间能够以最快的速度进行网络通信。HiperSocker提供了一个内部的虚拟局域网，各个节点通过TCP/IP协议通信。由于HiperSocket在内存中运行，不提供任何外部接口与外部网络连接，因此它保证了z系统能够将自己的网络资源完全用于外部通信，提高了系统的性能，同时保证了虚拟机间网络通信的安全性。

在操作系统级别， z系统配套的操作系统z/OS的Workload Manager（WLM）模块提供了工作负载虚拟化的功能，即在一个z/OS系统中运行

多个工作负载，并能够管理它们的优先级。Workload Manager根据用户的服务级别协定（Service Level Agreement，SLA）自动选择优化的方法，也就是说它能够将用户的业务逻辑转换为相应的资源优化操作。通过整合Workload Manager，用户在z系统上运行一个系统时，能够享受最大程度的便利。用户只需要提出自己应用的资源需求，发布应用，编辑应用的特征协议和对于SLA的需求，就不需要其他操作了。z系统负责为应用分配适合的逻辑分区、监控资源并根据SLA进行优化。

图4.2展示了z系统的架构，读者可以从中了解PR/SM、逻辑分区、HiperSocket、z/OS等技术在虚拟化架构中的层次位置。

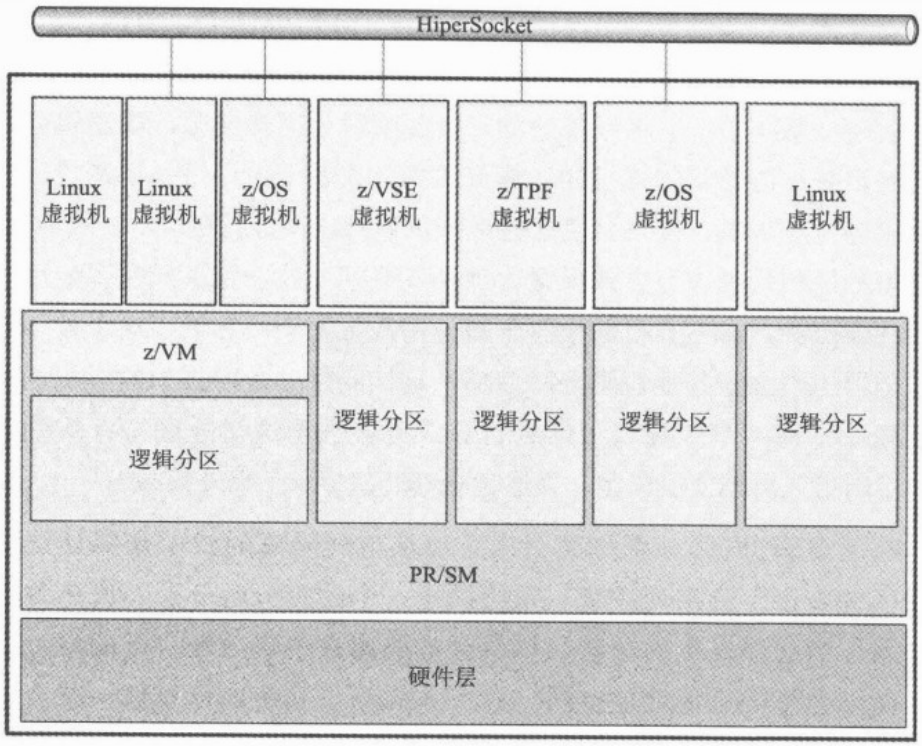



图4.2　IBM System z虚拟化架构

总之，IBM在System z系列主机上的虚拟化技术有几十年的经验，System z系列主机成为承载异构软件解决方案最理想的平台。在主机硬件层上，虚拟化提供的内核是z/VM，而虚拟机本身的操作系统可以来自其他厂商。z/VM的虚拟机资源管理器（Virtual Machine Resource Manager，VMRM）不仅能给虚拟机自动分配系统资源，也支持用户设定资源限制、资源优先和SLA。目前，最新型号的System z10服务器（如图4.3所示）能够同时运行上千台z/OS、z/Linux和TPF虚拟机。

图4.3　IBM System z10服务器

## 4.1.3 p系列服务器

为了满足中小型企业对IT基础设施和数据处理业务在性能和扩展性方面的要求，IBM公司推出了System p系列服务器，也就是常说的小型机，如图4.4所示。之所以称之为“System p”是因为它采用了IBM POWER处理器。它的前身是RS/6000，后来演化为IBM公司一套基于RISC架构和UNIX族操作系统的服务器和工作站产品线。System p目前支持AIX、Linux和UNIX操作系统。从POWER 4处理器发布起，IBM公司就开始从z系列服务器向p系列服务器引入虚拟化技术，此后所有的System p服务器都支持虚拟化技术。目前System p主要使用的是POWER 5处理器，正在逐渐过渡到POWER 6处理器。

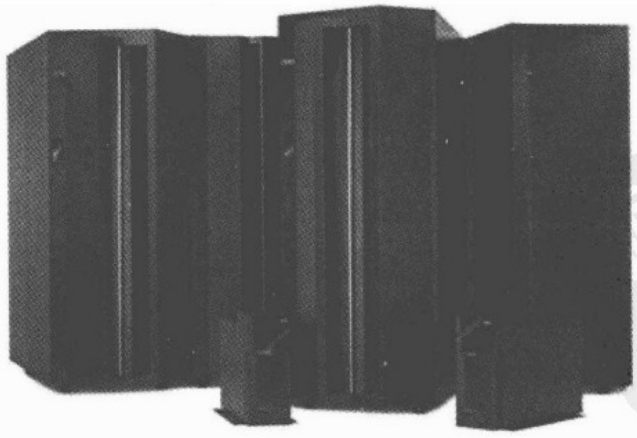


图4.4　IBM System p服务器

经过在虚拟化领域多年的探索和积累，IBM公司将p系列中的硬件、软件虚拟化功能整合到了PowerVM品牌下。PowerVM提供三个版本：快捷版、标准版和企业版。功能依次增强。快捷版包括了Power Hypervisor虚拟机监视器、共享专用容量（Shared Dedicated Capacity）、能够在Power系统下运行x86 Linux服务器的PowerVM Lx86、虚拟I/O服务器（Virtual I/O Server）、整合虚拟化管理器（Integrated Virtualization Manager，IVM）等功能，并支持在一台服务器上运行最多三个逻辑分区。标准版在快捷版的基础上，加入了在POWER 5系统上运行多个逻辑分区的支持，以及在POWER 6系统上的多个共享处理器池（Multiple Shared-Processor Pool）的支持。企业版只支持POWER 6系统，在标准版的基础上增加了实时分区迁移（Live Partition Mobility，LPM）功能。下面分别介绍PowerVM中涉及的关键技术。

Power Hypervisor与z系统中的PR/SM类似，是p系统虚拟化的基础。它将p系统中的硬件资源进行逻辑抽象，划分给各个逻辑分区，并保证逻辑分区之间的隔离。Power Hypervisor还负责根据逻辑分区的工作负载对资源进行调整，并提供多个分区之间的通信功能，这个功能还是虚拟I/O服务器的基础。

2001年，IBM公司在pSeries 690服务器中首次引入了在z系统上使用已久的逻辑分区技术和高可用性集群解决方案。2002年，又在逻辑分区的基础上开发了“动态逻辑分区”（Dynamic Logic Partition，DLPAR）技术，并发布了支持该技术的操作系统AIX5L V5.2，该操作系统运行在p系列服务器上。动态逻辑分区技术实现了硬件资源的按需分配，它可以在无需重启分区操作系统的情况下，动态分配CPU、内存和其他资源。这种动态分配资源的能力简化了对p系列服务器的管理工作，给用户提供了更大的灵活性。

2004年IBM公司在采用POWER 5处理器的p系列服务器中推出了微分区（Micro Partitioning）功能。微分区是一种芯片级的虚拟化，它使动态逻辑分区的资源调整功能不但能够调整物理资源，还可移动、增删虚拟资源。微分区允许多个分区共享一组物理处理器的计算能力，以1/10的物理处理器为单位为分区分配资源。POWER系统管理程序基于工作负载来调节分配给每个共享处理器分区的处理器数量。通过参数调优，系统管理员可以控制每个分区使用的处理器数量，过剩的处理能力可以分配给同一个共享处理器缓冲池中的其他分区。当工作负载发生变化的时候，微分区可以自动、平滑地调整分区资源。

2007年5月，IBM公司推出了主频高达4.7GHz的POWER 6处理器，以及基于POWER 6的System P570服务器。POWER 6处理器在虚拟化方面得

到了进一步的增强，每个芯片最多能被划分为1024个独立分区，每个分区都可以运行独立的操作系统和应用程序。

图4.5展示了一个典型的p系统中的Power Hypervisor、逻辑分区、微分区的架构。

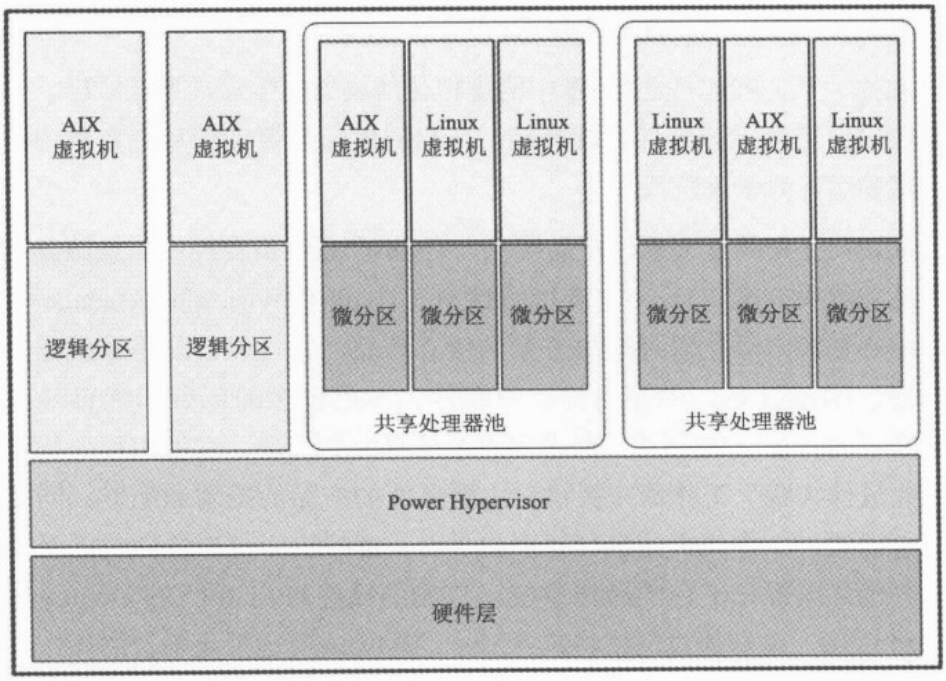



图4.5　IBM System p虚拟化架构

整合虚拟化管理器是POWER 5和POWER 6系统下的一个基于Web的虚拟化管理程序，用于管理小规模的p系统集群，提供基本的管理功能。整合虚拟化管理器的下层是虚拟I/O服务器。

虚拟I/O服务器是p系统中多个逻辑分区之间的通信模块，它提供了虚拟的存储接口—虚拟SCSI，以及虚拟的网络接口—虚拟以太网设施，能够支持p系统上绝大多数的存储设备和网络设备。

实时分区迁移（Live Partition Mobility，LPM）和实时应用迁移（Live Application Mobility，LAM）是POWER 6系统上独有的技术。实时分区迁移能够将正在运行的分区从一台物理服务器转移到另一台，并且避免计划中的和计划外的应用程序中断。这项技术大幅降低了单台设备宕机对整个系统的影响和迁移的成本，并且进一步提高了应用的可用性。实时应用迁移与工作负载分区（Workload Partition，WPAR）紧密相关。工作负载分区是基于AIX虚拟镜像创建的虚拟化操作系统环境。动态应用程序迁移是工作负载分区的一个重要组成部分，它允许用户在工作负载分区运行的时候，将工作

负载分区从一个逻辑分区移动到另一个逻辑分区。由于工作负载分区里的不同系统之间能够提供自动的、基于策略的工作负载重定位功能，它里面的应用不会受到影响，因此为工作负载分区提供了更高的可用性。

### 4.1.4 虚拟化管理

在这一节，我们按照第3章中的虚拟化解决方案生命周期的顺序，分别介绍IBM主流的虚拟器件管理产品。IBM的虚拟化管理产品主要集中在部署管理和运行时管理阶段。

在IBM公司的五大软件产品线里，Tivoli产品线负责提供信息服务管理的产品和解决方案。Tivoli家族里的TPM（Tivoli Provisioning Manager）是数据中心资源自动化管理解决方案的核心产品，它可以自动完成服务器、存储器、网络设备、操作系统、中间件、应用程序的部署和配置任务。TPM通过工作流（Workflow）来完成系统资源的部署，它使用预先构建的“行业最佳实践”工作流来提供对主要厂商的产品的控制和配置。同样，TPM可以通过工作流方式进行虚拟化平台、虚拟机、虚拟器件的部署。对于部署包含虚拟化平台的操作系统，TPM的插件TPM for OS Deployment（TPM OSD）能够通过网络将包含Xen、VMware等虚拟化平台的操作系统镜像部署到裸机上。目前，TPM OSD支持的操作系统主要有Linux、AIX、Solaris和Windows。对于虚拟器件部署，TPM利用开放流程自动库（Tivoli Open Process Automation Library，OPAL）实现了多个虚拟化器件跨多个虚拟化平台的自动化部署，并且避免了部署过程中可能出现的人为错误，大大提高了工作效率，如图4.6所示。通过扩展工作流，TPM还能够实现其他管理功能，例如安装系统更新等。

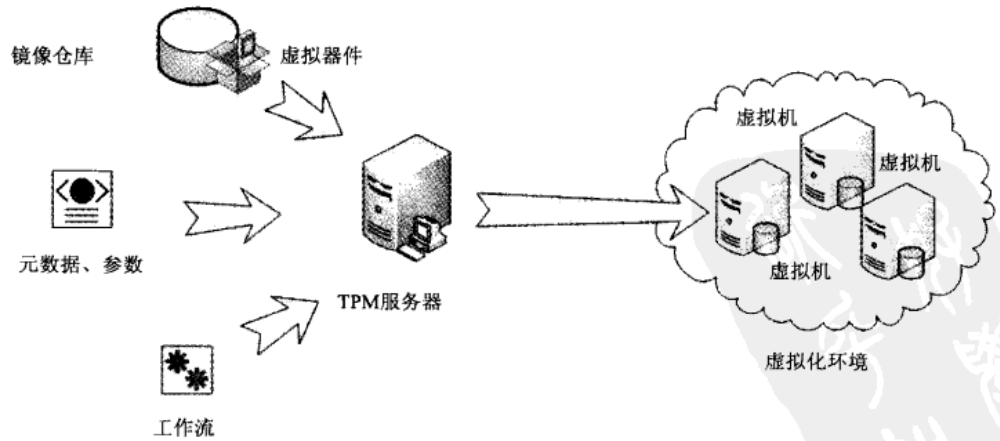



图4.6 TPM部署虚拟器件

部署阶段的另一个关键任务是配置虚拟器件。在这方面，IBM Tivoli软件提供了配置管理软件TADDM（Tivoli Application Dependency Discovery Manager）和CCMDB（Change and Configuration Management DataBase）。TADDM负责发现解决方案中的配置点及配置点之间的依赖关系，跟踪配置点的状态改变，并将用户的业务逻辑映射到对应的配置点上。CCMDB负责以数据库的方式存储数据中心的各种配置信息，例如存储、网络、服务器、虚拟机、应用和安全等。CCMDB还能够通过标准化的接口，协助服务发现、工作流管理及策略管理等工作。

在运行时管理阶段，最主要的两个工作是监控和操作。IBM在数据中心资源监控方面的产品是ITM（IBM Tivoli Monitoring）服务器。该产品能够提供统一的解决方案来监控数据中心中的所有关键资源，检测瓶颈和潜在的问题，在严重的情况下进行自动恢复，不需要系统管理员手动解决问题。ITM提供了对大部分主流硬件平台和操作系统的支持，用户可以通过ITM统一管理数据中心不同厂商的产品。针对虚拟化监控的特殊环境，ITM的扩展模块ITM for Virtual Server扩展了ITM端到端的监控和管理功能，能够监视Citrix Agent、VMware ESX和 Microsoft Virtual Server上虚拟机内应用程序的关键资源和流程的可用性，收集数据以便在IBM其他产品中同时显示实时和历史数据，从而进行深入诊断。该产品还有内置的专家知识库，IT运营部门无需邀请业务专家就可以处理更多事件。通过使用ITM，IT运营部门和管理员可以更准确地确定虚拟化服务的工作负载。图4.7是ITM监控虚拟化环境的示意图。

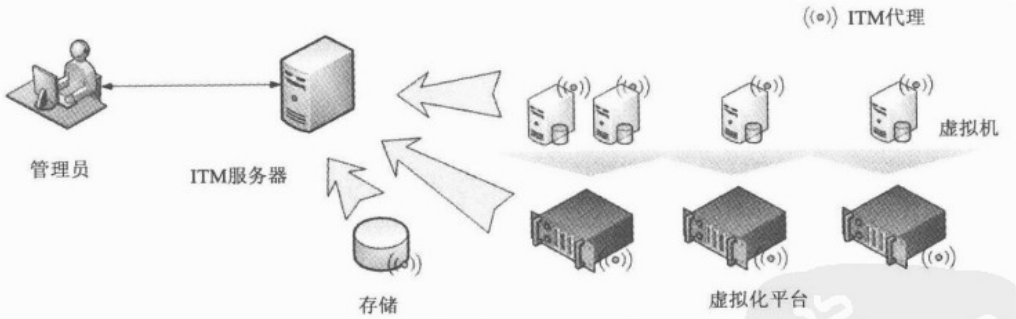


图4.7　ITM监控虚拟化环境示意图

在运行时管理的操作方面，IBM提供了Systems Director系列产品进行企业级系统管理操作。通过使用IBM Systems Director，IT运营部门能够更好地管理数据中心里所有的物理资源和虚拟资源。通过与Tivoli产品配合使用，该系列产品能够提供完整的跨企业服务管理。IBM Systems Director的主要功

能有：远程部署系统、查看和跟踪远程系统的硬件配置详情、监控关键组件的使用情况和性能、优化服务器的性能和可用性、对系统进行分析和故障诊断、维护和更新软件等。为了更好地支持对虚拟化环境的管理，IBM Director系列发布了IBM Virtualization Manager软件，用户通过使用该软件的控制台，不仅可以管理IBM z和p系列服务器上的虚拟化环境，还可以管理VMware ESX Server和Microsoft Virtual Server等虚拟化环境，从而大大减少了支持多种类型虚拟化平台所需管理工具的数量。该产品还提供了虚拟器件镜像的管理功能组件，辅助用户进行虚拟器件的部署。另外，它还集成了VMware等虚拟化管理工具的高级功能，使用户能够无缝地整合VMware产品和IBM的解决方案。

## 4.2 VMware

### 4.2.1 概述

1998年成立的VMware公司将大型机所特有的虚拟化技术带入到了基于x86架构的普通个人电脑领域。目前，该公司已经拥有x86虚拟化市场的较大份额。VMware的产品线可以帮助客户实现虚拟化基础设施、整合资源、提高资源利用率，在降低运营维护成本的同时，增强业务的灵活性、可用性和安全性。

2004年，VMware公司被EMC公司收购，成为EMC公司旗下一个独立的软件子公司。另外，从2004年起，VMware公司每年都举办一次VMworld大会，随着大家对虚拟化的日益关注，与会人数逐年递增。

确立了自己在x86架构上虚拟化平台提供商的地位以后，VMware公司又调整了战略计划，目标是整合虚拟化数据中心的基础设施，提供基于虚拟化基础架构的数据中心操作系统（Virtual DataCenter Operating System，VDC-OS）。这里的数据中心操作系统和操作系统的概念完全不同，它集成了数据中心所有的硬件资源、虚拟服务器和其他基础设施，通过有效的管理为上层应用提供可用、可伸缩、灵活的基础设施平台。在围绕这个目标进行新一轮的调整之后，VMware公司已经拥有了三条虚拟化产品线：数据中心产品、桌面产品和其他虚拟化辅助产品，它们涵盖了服务器虚拟化的整个生命周期。下面，我们将分别介绍这些产品线。

### 4.2.2 数据中心虚拟化

VMware的数据中心产品主要面向企业服务器市场，包括VMware Infrastructure、VMware vCenter Server系列管理软件、VMware Capacity Planer、VMware Data Recovery和VMware Server等。

VMware Infrastructure是一个功能丰富的虚拟化软件套件，能够提供虚拟化基础架构、应用程序和管理等多种服务，主要组件包括ESX Server/ESXi Server、VMFS、Virtual SMP、DRS、VMotion、Storage VMotion、HA、Consolidated Backup、VirtualCenter Agent。

ESX Server是数据中心虚拟化的基础，它能够整合数据中心的计算资源、网络资源和存储资源，并将它们动态地分配给虚拟机。早在2001年，VMware公司就推出了面向企业用户的VMware ESX 1.0（代号Elastic Sky X）。ESX经过多年的发展，成为VMware公司最重要的企业级虚拟化平台产品，也是虚拟化软件套件VMware Infrastructure中最重要的组成部分。从2003年开始，ESX支持了虚拟对称处理器（virtual Symmetrical Multi-Processing，vSMP），这给ESX的性能带来了很大程度的提高，有利于在ESX上部署对计算资源要求甚高的企业级应用，比如ERP和CRM应用。

VMware ESXi是VMware公司于2008年推出的最新的免费虚拟化平台，在保持ESX Server功能的前提下，它对原有的虚拟化平台进行了大幅裁剪，仅需要32MB磁盘空间，这使得ESXi的安全性有所提高，成为“固件”虚拟化平台合适的选择。ESXi上所运行的虚拟机性能接近于物理机的性能。和VMware Infrastructure整合后，用户可以在ESXi上使用服务器整合和自动负载平衡的功能。

除了ESX Server和ESXi Server，VMware Infrastructure还包括了能够给虚拟机及其上层应用提供可用性、可扩展性和安全性的高级功能。VMFS是一种高性能的文件集群系统，通过它，多个ESX Server可以访问同一个存储。VMotion是VMware公司实现的实时迁移技术，它可以把在一台物理机上运行的虚拟机迁移到与其共享同一个存储的另一台物理机上。Storage VMotion允许把在一台物理机上运行的虚拟机迁移到非共享同一个存储的另一台物理机上。DRS利用VMotion技术动态平衡同一个资源池内所有虚拟机的资源，动态地满足因虚拟机负载变化引起的资源需求。HA可以防止物理机故障对虚拟机产生的影响，当检测到物理机故障时，HA利用VMotion技术将故障物

理机上的虚拟机迁移到其他物理机上。Consolidated Backup为虚拟机提供了集中型备份工具。vCenter Agent可使VMware Infrastructure与vCenter连接，使VMware Infrastructure成为可管理、可配置的虚拟化平台。

VMware vCenter系列解决方案是一个可扩展的虚拟化平台管理工具集，使用户能够对数据中心中数量庞大的物理机和虚拟机进行集成管理。该系列解决方案以vCenter Server为核心。vCenter Server通过vCenter Agent与VMware Infrastructure中的ESX Server连接，数据中心管理员能够通过vCenter Server提供的统一管理控制台，快速部署虚拟机并监控物理机和虚拟机的性能，集中优化管理VMware Infrastructure环境。此外，VMware公司还提供了其他可与vCenter Server集成的产品，包括vCenter Site Recovery Manager、vCenter Lab Manager、vCenter Lifecycle Manager、vCenter Stage Manager、vCenter AppSpeed等，从而提供其他高级功能。下面分别介绍VMware vCenter和这些与之集成的产品。

2003年，VMware公司推出了能够集成多个ESX虚拟化平台的管理工具VMware VirtualCenter，就是vCenter的前身。经过多年的发展，VirtualCenter不断增加新的管理特性，已经成为VMware公司虚拟化战略中不可或缺的管理工具。在最新的产品线调整中，VirtualCenter被更名为vCenter。vCenter Server主要提供三个功能：虚拟机的部署和迁移、虚拟化平台和虚拟机的管理、系统监控。虚拟机的部署和迁移包括了集成的从物理机到虚拟机的转换、虚拟机克隆、实时迁移及虚拟机磁盘的实时迁移，使得虚拟机能够部署到虚拟化平台上，并且在各个同构的虚拟化平台之间移动。用户可以通过vCenter Server的客户端或者Web方式远程接入vCenter Server进行各种操作。另外，用户通过vCenter Server可以从单个界面持续监控物理服务器和虚拟机的可用性和利用率，其中的性能曲线图可以帮助用户分析虚拟机、资源池及服务器的利用率和可用性。最后，vCenter Server能够生成报告供用户做离线分析。一旦发生异常情况，vCenter Server还能向用户发出警报和通知。

vCenter Site Recovery Manager提供了与虚拟化数据中心灾难恢复有关的自动化管理和执行功能，从而帮助用户简化恢复流程，降低恢复风险。它主要包括了三大功能：灾难恢复管理、无中断测试和自动化故障切换。灾难恢复管理功能通过与vCenter Server的整合，可以直接在vCenter Server上运行与操作，用户也可以通过自定义脚本来扩展恢复计划，使恢复过程具有一定的灵活性。无中断测试可以帮助用户在不影响复制数据的情况下自动测试恢复计划，从而保证了恢复的可行性。自动化故障切换允许用户

暂停恢复过程，并且对参数进行重新配置。

vCenter Lab Manager创建和管理共享的虚拟机镜像库。用户只需要从该镜像库中选择需要部署的镜像并进行一些简单的操作，vCenter Lab Manager就可以按需进行动态部署。vCenter Lab Manager保证部署的实例之间不存在资源冲突，并且还能够定义用户的角色和访问权限。

vCenter Lifecycle Manager按照不同的用户角色对数据中心内虚拟机的生命周期进行管理。这些角色包括了普通用户、审批者、IT员工和IT管理员，他们在虚拟机的生命周期内有交互，vCenter Lifecycle Manager通过规范的流程将这些角色关联起来，实现了对虚拟机从始至终的有效管理。

vCenter Stage Manager负责部署和更新虚拟化服务。它的主要功能有可视化虚拟化服务、在各个阶段之间轻松转换服务配置、按服务进行访问控制、优化资源利用率等。2009年7月，VMware公司宣布把vCenter Stage Manager合并到vCenter Lab Manager中。

vCenter AppSpeed通过主动监测虚拟化服务性能的变化，发现性能瓶颈，帮助调整分配给虚拟化服务的相关资源，从而在工作负载动态变化的情况下满足服务级别协定（SLA）要求。

VMware vCenter Converter是一款物理机—虚拟机转换（P2V）软件。它可以通过管理控制台和转换向导，在较短的时间内将安装有Microsoft Windows操作系统的物理机转换为VMware格式的虚拟机。另外，它还可以在两个不同的VMware平台之间进行虚拟机转换。

VMware Server是VMware公司提供的免费服务器虚拟机监视器。与ESX Server不同，VMware Server需要作为一个应用程序安装在Windows或Linux操作系统上，而虚拟机则运行在VMware Server上。由于没有直接安装在物理机上，因此VMware Server的性能不如ESX Server。VMware Server的前身是VMware GSX Server，早在2001年就与VMware ESX Server一同推出。与ESX的命运不同，VMware公司2006年的战略调整中将GSX Server更名为VMware Server并免费提供给用户使用。

VMware Capacity Planner是商业及IT资源规划工具，提供经过整合的分析、规划和决策支持功能，使得基础架构评估服务更快速、更精确及更容易测量。

### 4.2.3 桌面和应用虚拟化

桌面产品面向企业桌面用户或者个人用户，包括VMware View、

VMware Workstation、VMware Fusion、VMware ThinApp和VMware ACE。

VMware View是VMware公司的虚拟桌面产品。该产品在数据中心集中保存了所有用户的个性化虚拟桌面，然后通过网络向用户的终端设备提供虚拟桌面。对于数据中心的管理人员来讲，VMware View以集中管理的方式维护用户的虚拟桌面，简化了管理复杂度，也更加灵活有效；而对于用户来讲，可以在有网络的任何地方访问自己的个性化桌面，而不受某个具体终端设备的限制。

VMware Workstation允许在单台个人电脑上同时运行多个操作系统，包括Windows和Linux。它可以挖掘个人电脑在性能方面的潜力，提高其资源的利用率。1999年，VMware公司推出了能够在Windows和Linux上运行的VMware Workstation 1.0，实现了在单台个人电脑上同时运行多个操作系统。从2001年到2008年，该产品几乎每年都升级一个版本，成为VMware公司初期的重要产品之一。后来，VMware公司开始把研发重心转移到企业级虚拟化产品上，VMware Workstation的升级进度受到了一定的影响。到2008年，VMware Workstation的版本是6.5。

VMware Fusion专为Mac平台设计，它可使用户在基于Intel架构的Mac操作系统上运行Windows程序。VMware Fusion通过一系列技术，尽可能保证Window程序的安全运行，它还实现了Windows应用和Mac应用的数据共享。

VMware ThinApp是一款应用程序虚拟化产品，可以实现在同一操作系统上运行多个虚拟应用程序而不发生冲突，甚至可以同时运行同一应用程序的不同版本。ThinApp的应用程序虚拟化技术使得应用程序独立于操作系统和其他应用程序，封装后的应用程序无需进行回归测试，避免了与其他应用程序可能发生的冲突。ThinApp通过共享网络驱动器的方式对虚拟化应用程序进行透明的流式传输，加快了软件的部署速度。同时，它还可以方便地对虚拟化应用程序进行升级。

VMware ACE是一个面向企业的解决方案，它把每个员工所需的操作系统和所有应用分别打包在被称为ACE的虚拟机中，然后将这些ACE放在数据中心里进行集中管理。这样，员工通过网络远程访问自己的ACE，管理员则通过动态策略来控制员工访问设备和网络的权限。由于员工所涉及的数据不在本地，它最大限度地保证了公司的机密数据不被外泄。

### 4.2.4 虚拟化辅助工具

虚拟化辅助工具包括其他一些实用的虚拟化工具，如VMware

VMmark、VMware Player和VMware Studio。

VMware VMmark是x86架构下的虚拟化标准测试工具，它可以对运行在虚拟化环境中的应用程序进行性能测量。无论对于设备制造商、软件供应商，还是对系统集成商，VMmark都是一款有用的工具。它既可以测试虚拟机的性能，也可以测试构建在不同硬件上的虚拟化平台的性能，从而帮助准备实施虚拟化的企业做硬件采购决策。

VMware Player是一款免费的运行在Windows和Linux上的虚拟化软件应用程序。虽然它本身不能创建和管理虚拟机，但是它能够运行多种虚拟机，这些虚拟机可以来自VMware Workstation、VMware Fusion、VMware Server或VMware ESX。另外，VMware Player也具备在主机和虚拟机之间共享数据的功能。

VMware Studio是定制虚拟化镜像的工具。用户通过VMware Studio基于网页的控制台，可以创建支持开放虚拟格式（Open Virtualization Format，OVF）的定制化虚拟镜像。此外，VMware Studio还可以为已经部署了的OVF镜像包提供自动更新。

## 4.3 Xen/Citrix

### 4.3.1 概述

Xen是一款开放源代码的虚拟机监视器，最早由剑桥大学开发，在x86、x86_64、PowerPC和其他CPU架构上都能提供强大、高效和安全的虚拟化特性。Xen能够支持广泛的客户操作系统，包括Windows和Linux的多种发布版本。

2003年，Xen的研究人员就在SOSP会议上发表了名为“Xen and the Art of Virtualization”的论文，这篇论文描述了Xen 1.x的架构，并首次提出了半虚拟化的概念。2004年在FREENIX会议上，Clarkson大学的研究人员发表了名为“Xen and the Art of Repeated Research”的论文，独立验证了前一篇论文的结果。

也是在2003年，在Xen开源社区的支持下，面向企业用户的小型商业公司XenSource成立了。该公司基于Xen平台向企业用户提供虚拟管理软件。2005年，XenSource公司发布了Xen V3，成为Xen第一个真正能够满足企业需

求的版本。这个版本的Xen能够运行在多种32位处理器上，并且支持Intel的VT技术。Xen V3也支持物理地址扩展（Physical Address Extension，PAE），从而支持在32位机器上内存大于4GB的情况。但是，此时的Xen仍然只支持Linux作为客户操作系统。由于V3的发布，XenSource公司第一次发布了提供给企业的解决方案XenOptimizer。XenOptimizer在Xen上提供管理接口，使用户能够管理和控制虚拟环境下的各种资源。

在2006年下半年，XenSource公司发布了XenEnterprise 3.0。该产品成为VMware公司的直接竞争对手。在Xen V3.03的基础上，XenOptimizer包含了新的管理和监控界面。最重要的是，它已经能够支持Windows作为虚拟机操作系统。这归功于在2006年7月XenSource公司和Microsoft公司达成的一个伙伴协议，他们联合开发代号为Viridian的虚拟机监视器，这也是Microsoft Hyper-V的前身。

思杰（Citrix）公司是一家全球知名的应用交付基础架构解决方案提供商。2007年，思杰公司收购XenSource公司，进入服务器虚拟化市场。思杰的虚拟化解决方案都是基于开源虚拟化平台Xen构建的。在思杰收购了XenSource公司以后，将其服务器产品改名为XenServer，将Xen开源社区移到xen.org，并通过对XenSource和已有技术的整合，提出了“交付中心”的概念，其中包括服务器虚拟化、应用虚拟化、桌面虚拟化三条产品线，如图4.8所示。整合以后，思杰公司已经能够在这些领域为客户提供全线解决方案，成为全面的虚拟化技术和服务的提供商。

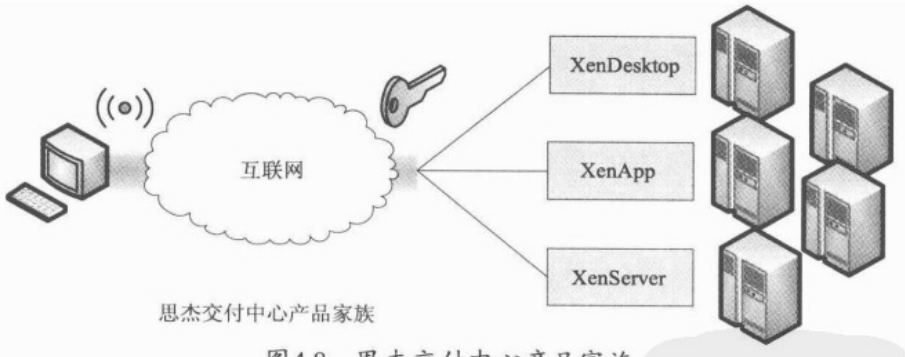



图4.8　思杰交付中心产品家族

从战略角度来看，当初思杰公司的强项仅限于应用虚拟化，但是在收购了Netscaler和XenSource等公司以后，其虚拟化的战略也逐渐明晰起来，这就是“将数据中心变为交付中心”。它的交付中心向用户提供的是一个全面的应用交付方案，可以将运行在数据中心中的应用和定制化配置高效而安全地交付给用户。虚拟化是整个交付中心的核心技术，从服务器虚拟化，到应

用虚拟化，再到桌面虚拟化，用户所体验到的是一个端到端的虚拟化解决方案。现在，思杰公司已经成为虚拟化市场中不可忽视的重要力量。

### 4.3.2 服务器虚拟化

早期以VMware公司为代表的全虚拟化是通过二进制转换来实现的，然而二进制转换的性能代价相对较高，一次转换会牺牲几百甚至上千个CPU时钟周期。Xen的研究人员通过修改虚拟机的操作系统，使得虚拟机操作系统能够在需要调用物理资源时向底层的虚拟机监视器发出超级调用（hypercall）指令。这种方式消耗的CPU时钟周期远远小于二进制转换的方式，因此性能有较大的提升。

正是由于Xen的半虚拟化模式要修改虚拟机操作系统，Xen的服务器虚拟化具有一定的局限性，即不能支持所有的操作系统。比如，它在该模式下不能支持Windows操作系统，对Linux版本有特殊的限制。但是不可否认的是，高性能的Xen虚拟化技术的出现，突破了VMware一家独大的格局，给x86虚拟化市场带来了新鲜的血液，给客户带来了新的选择。

随着Intel和AMD两大厂商相继推出了x86架构下支持虚拟化技术的CPU，Xen可以在这种架构的CPU上实现全虚拟化，能够支持包括Windows在内的多种操作系统。另外，思杰公司收购Xen使得Xen从真正意义上成为了能够支持企业级应用的虚拟化平台。

### 4.3.3 应用虚拟化

XenApp是思杰公司的应用虚拟化产品。相对于其他虚拟化而言，应用虚拟化是思杰公司最早涉足的领域，起初叫Presentation Server，后来改名叫XenApp。XenApp作为Windows应用的交付系统，主要负责管理数据中心的所有应用，以实现良好的应用性能、灵活的应用交付和安全的信息交互。

具体而言，XenApp通过两种方式将应用交付给用户，一种是将应用集中运行在数据中心里，用户在本地设备上通过网络访问该应用；另一种是将应用通过流方式交付到用户设备运行。XenApp能支持多种用户设备，使用PC的办公用户、需要对业务进行批处理的用户及使用移动设备的用户都可以用XenApp。XenApp在数据中心管理应用和用户数据，不受用户端的影响。这种应用交互模式给用户带来了很大的好处。首先，由于应用和数

据都不在用户的本地设备里，免去了安装和升级应用的麻烦。其次，在用户本地设备受到病毒和木马侵害时，这种模式可以保证数据仍然安全。最后，应用在数据中心，管理变得更容易，降低了维护成本。即使某个应用出现了问题，管理员也可以通过XenApp远程操作功能对该应用进行维护。

### 4.3.4 桌面虚拟化

人们喜欢按照自己的喜好在终端设备上设置桌面，比如在Windows桌面设置各种快捷方式。但是，在一般情况下，这种桌面设置只在当前的终端设备下生效，如果换一台终端设备，桌面环境就改变了。桌面虚拟化可以解决这个难题，它可以灵活地交付和管理用户桌面，并且满足用户的各种桌面需求。而今，随着虚拟化桌面技术的广泛采用，企业希望拥有高性能、个性化的桌面虚拟化解决方案，用户也希望使用虚拟化桌面获得与本地桌面相媲美的用户体验。

思杰公司提供基于服务器端的桌面虚拟化，也称为终端虚拟化，其对应的产品是XenDesktop。它在数据中心的服务器端构建一个虚拟桌面架构（Virtual Desktop Infrastructure，VDI），只要用户通过XenDesktop设置了自定义桌面，那么在支持XenDesktop的任意一台终端设备上都能随时随地通过一定的网络协议（如远程桌面）访问服务器端的个性化桌面系统。XenDesktop需要和XenApp和XenServer配合使用。XenServer作为服务器虚拟化平台，用户的虚拟桌面工作负载运行于XenServer之上，通过XenApp向用户交付应用，通过XenDesktop将虚拟化桌面呈现给用户。

通过XenDesktop、XenApp和XenServer的结合，思杰公司为用户提供了一套完整的桌面虚拟化解决方案。但思杰公司并不满足于此，为了提高用户体验，思杰公司还在网络传输、应用的可用性、性能优化和安全性等方面下足了工夫。XenDesktop 3采用了CitrixHDX媒体流技术，把经过压缩的流媒体内容发送至用户端，并在用户端设备上播放，从而改善了用户的多媒体体验。XenDesktop 3还采用CitrixHDX即插即用技术，对本地设备提供了透明化的支持。

## 4.4 Microsoft

### 4.4.1 概述

微软（Microsoft）公司长期以来都是桌面操作系统及办公软件的重要提

供商。2008年，微软公司推出Windows Server 2008和Hyper-V，进入服务器虚拟化市场。在虚拟化战略上，微软公司非常重视服务器虚拟化、应用虚拟化、桌面虚拟化及虚拟化管理产品。微软公司的虚拟化产品线布局如下：以Virtual Server和Hyper-V为代表的服务器虚拟化，以Application Virtualization为代表的应用虚拟化，以VDI为代表的桌面虚拟化和以System Center为代表的虚拟化管理软件。微软公司的目标是实现“从数据中心到桌面”的虚拟化战略。下面，我们简单回顾一下微软公司的虚拟化技术发展历程。

微软公司于2003年收购了做Virtual PC软件的Connectix公司，并于2003年底推出了Microsoft Virtual PC，将Virtual PC用于培训课程，可使用户方便地在多个不同的操作系统环境间切换。

2005年，微软公司推出了自己的第一款虚拟化产品Microsoft Virtual Server 2005，并提供基于Virtual Server的管理工具。

2006年，微软公司宣布Virtual PC和Virtual Server 2005 R2企业版为免费软件。同年，其竞争对手VMware公司也不约而同地将VMware Server 1.0宣布为免费软件。

2008年，微软公司进军虚拟化市场的利器Hyper-V问世。微软公司整合了虚拟化产品线，包括服务器虚拟化、应用虚拟化、桌面虚拟化和虚拟化管理软件，成为虚拟化市场中有力的竞争者，如图4.9所示。

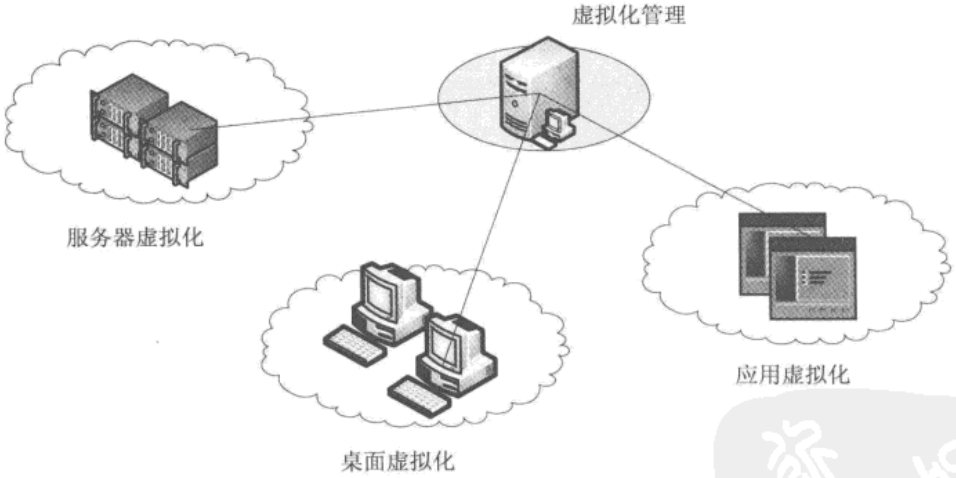



图4.9　微软公司的虚拟化产品

虽然微软公司进入虚拟化市场的时间并不长，但是已经引起了极大的反响，一方面是因为微软公司有着庞大的用户群，另一方面也归功于微软公司提供的解决方案非常全面。从服务器虚拟化到应用虚拟化，再到桌

面虚拟化，微软提供给用户的是一个端到端的产品集，用它们来创建、配置、部署和管理虚拟化数据中心的物理服务器、虚拟机、存储、网络及最上层的应用程序。这些全面和强大的功能集可以通过统一的、集成的界面来进行监控和操作，既降低了用户的成本，减少了复杂性，又保证了应用的灵活性和可用性。

在接下来的几个小节里，我们将介绍微软公司的虚拟化产品和相关技术。

### 4.4.2 服务器虚拟化

微软公司在服务器虚拟化领域拥有两款产品，分别是Windows Virtual Server 2005 R2和Windows Server 2008的Hyper-V。

Windows Virtual Server 2005 R2是微软公司较早推出的企业级虚拟化平台，类似于VMware公司的VMware Server。它运行在Windows Server 2003上，通过打补丁的方式进行安装，使得宿主机上能够运行多个客户Windows操作系统。由于该产品属于应用程序，因此需要安装在已有的Windows操作系统上，由已有的操作系统负责分配资源，Virtual Server再将这些资源分配给运行的虚拟机。从实现虚拟化的层次来讲，它们处于操作系统之上，需要和已有操作系统配合才能实现虚拟化平台。

随着微软公司推出Windows Server 2008，微软新一代的虚拟化平台Hyper-V也强势推出。Hyper-V是一个更灵活、更健壮、性能更强的虚拟化平台。由于和Xen在虚拟化技术上的长期深入合作，Hyper-V从一定程度上借鉴了Xen的设计思想和架构，并且在微内核架构上充分利用了Windows操作系统经典的驱动模型，将Hyper-V本身大小控制在300KB左右。不同于之前推出的Virtual Server虚拟化平台，Hyper-V实现虚拟化的层次是在操作系统级别，在操作系统核心直接参与虚拟硬件资源的分配和调度，所以性能有了很大的提高。不过，Hyper-V对硬件和操作系统的要求也比较高，处理器必须是支持虚拟化技术的Intel VT和AMD V，操作系统必须是64位的Windows Server 2008。

### 4.4.3 应用虚拟化

和很多公司一样，微软公司也在关注应用虚拟化。我们知道，如果操作系统和应用程序不兼容，可以用服务器虚拟化来解决；而如果同一操作

系统上的应用程序不兼容，则需要用到应用虚拟化来解决。微软公司的应用虚拟化产品叫做Microsoft Application Virtualization，简称App-V。

App-V主要分为三个部分：虚拟应用服务器、虚拟应用客户端和应用序列化器。虚拟应用服务器负责集中存储和管理生成的虚拟应用和服务；虚拟应用客户端负责在用户终端创建虚拟应用的运行环境，它先从服务器端获得应用，然后将应用缓存在本地，接着虚拟应用就可以在本地运行了；应用序列化器负责将常规应用序列化，生成虚拟应用。由于App-V的虚拟应用不需要安装，因此用户可以通过流形式从服务器端按需、实时地获得应用，大大缩短了部署应用和服务的时间。

### 4.4.4 桌面虚拟化

微软公司的桌面虚拟化产品有两类：基于客户端的桌面虚拟化和基于服务器端的桌面虚拟化。

在客户端实现桌面虚拟化的方式，本质上与服务器虚拟化没有差别，只是构建虚拟化平台的物理机从数据中心的服务器变成了本地PC。一般来说，与服务器上的虚拟机监视器相比，PC上的虚拟机监视器功能有所减少，性能有所降低。Virtual PC是微软公司在客户端桌面虚拟化领域推出的产品，前身是Connectix公司的Virtual PC，2003年被微软公司收购。

客户端实现桌面虚拟化的一大用途是在本地解决应用和操作系统不兼容的问题。比如本地操作系统是Windows Vista，用户需要运行一个Windows XP SP2上的应用程序。这时，用户可以在本地安装Virtual PC，并通过Virtual PC安装一个Windows XP SP2客户操作系统，将所需应用直接安装在客户操作系统中，就可以在本地使用该应用了。

在服务器端的桌面虚拟化领域，微软公司也推出了两款产品：Terminal Services Remote Desktop和Microsoft Virtual Desktop Infrastructure。

在微软公司以前的虚拟化产品划分中，Terminal Services Remote Desktop（TSRD）属于表示层虚拟化（Presentation Virtualization），在最近的调整中被归为桌面虚拟化的一种。TSRD其实是实现了会话（Session）的虚拟化，即在同一个服务器上保存不同用户的会话，而会话之间不会产生冲突，这样用户可以在任何一个终端设备上通过网络随时访问自己在服务器上的会话。

Microsoft Virtual Desktop Infrastructure（MVDI）与其他虚拟化厂商提供的VDI类似，也是利用服务器虚拟化技术，在物理服务器上运行多个虚拟机，然后在每个虚拟机上创建多个桌面系统，用户可以通过远程桌面程序从本地连接到服务器端的桌面。

### 4.4.5 虚拟化管理

如果仅有前几小节所提到的虚拟化技术，而没有虚拟化管理软件的支持，微软公司的虚拟化战略就少了很重要的一环。随着Windows Server 2008的推出，微软公司也推出了强大的虚拟化集成管理软件System Center，它可以精简操作，从而降低管理复杂性。System Center是一个系统管理产品家族，主要产品包括Virtual Machine Manager、Data Protection Manager、Operation Manager和Configuration Manage。其中，Virtual Machine Manager（简称SCVMM）是专用于数据中心虚拟机管理的软件产品。

SCVMM能够管理数据中心的基础架构，包括数据中心的服务器，以及在其上运行的虚拟机。它可以动态优化这些平台的虚拟化资源，提高资源利用率。SCVMM能够管理多个虚拟化平台，既可以管理VMware ESX服务器，也可以管理微软Windows Server 2008 Hyper-V。对于用户来说，它能够实现端到端的解决方案，包括规划、部署、管理和优化。

## 4.5 小结

本章对虚拟化业界四家主要厂商的基本情况和主要产品进行了介绍。

IBM公司提供了业界最广泛的虚拟化能力。采用跨平台的虚拟化、自动化和系统管理解决方案，用户能够简单、动态地访问和管理资源，以实现更高的资产使用率和更低的运行成本。IBM公司在大型机和小型机的高端虚拟化市场占据了领先地位，同时还提供了业界领先的虚拟化领域的产品解决方案来帮助数据中心实现虚拟化部署、监控和管理自动化。

VMware公司是最早进入x86虚拟化领域的厂商，占据了x86虚拟化市场较大的份额。通过对产品线的调整，VMware公司明确了面向数据中心的产品、面向桌面的产品和其他辅助工具的产品线定位。VMware公司的主要精力仍然集中在数据中心服务器虚拟化这一领域，这符合VMware公司提出的“构建虚拟化数据中心”的战略思想，为未来云计算的发展打好基础。

Xen是开源的虚拟化厂商，通过x86架构下半虚拟化技术的创新，Xen为虚拟化开辟了新的道路。2007年，思杰公司收购了Xen，经过一段时间的整合后，思杰公司已经成为x86服务器虚拟化市场一支不可忽视的力量。我们从服务器虚拟化、应用虚拟化和桌面虚拟化三个方面介绍了思杰公司的产品，“将数据中心变为交付中心”这样的理念也让读者清晰地了解了思杰公司的战略布局。

微软公司凭借强大的软件研发实力，在x86虚拟化市场获得了一席之地，产品线也较为全面。我们介绍了微软服务器虚拟化、应用虚拟化、桌面虚拟化和虚拟化管理四个方面的产品，并且解析了微软公司“从数据中心到桌面虚拟化”的战略构想。

CHAPTER 5

# 第5章 云计算概论

从20世纪40年代世界上第一台电子计算机诞生至今已经过去了半个多世纪。在这几十年里，计算模式经历了单机、终端—主机、客户端—服务器几个重要时代，发生了翻天覆地的变化。半导体芯片技术遵循着摩尔定律不断的发展，到2009年世界上最快的计算机IBM Roadrunner已经达到每秒千万亿次的运算速度。在过去的二十年里，互联网将全世界的企业与个人连接了起来，并深刻地影响着每个企业的业务运作及每个人的日常生活。用户对互联网内容的贡献空前增加，软件更多地以服务的形式通过互联网被发布和访问，而这些网络服务需要海量的存储和计算能力来满足日益增长的业务需求。

互联网使得人们对软件的认识和使用模式发生了潜移默化的改变。计算模式的变革必将会带来一系列的挑战。如何获取海量的存储和计算资源？如何在互联网这个无所不包的平台上更经济地运营服务？如何才能使互联网服务更加敏捷、更随需应变？如何让企业和个人用户更加方便、透彻地理解与运用层出不穷的服务？“云计算”正是顺应这个时代大潮而诞生的信息技术理念。目前，无论是信息产业的行业巨头还是新兴科技公司，无不把云计算作为企业发展战略中的重要组成部分。云计算的号角已经吹响，势不可挡。本章将解释云计算的确切含义与分类，分析云计算的优势与其带来的变革，并阐述云计算的来龙去脉。

## 5.1 云计算的概念

“云计算”这个词相对于“分布式计算”或“网格计算”等技术类名词的确显得更加浪漫，甚至很难让人们从这个词本身推断它所涵盖的范畴。事实上，不但第一次听说“云计算”的普通技术工作者会感到不知所

云，就连众多行业精英和学术专家们也很难为云计算给出一个准确的定义，每个人从不同的角度会有不同的解释。本节将首先呈现云计算的四个典型案例，并以这些案例为脉络，探究云计算的内涵，领略云中的真实世界。

## 5.1.1 走近云计算

### 1. 相关案例

[案例一] 2008年3月19日上午10点，美国国家档案馆公开了希拉里克林顿在1993～2001年作为第一夫人期间的白宫日程档案。由于这些档案是新闻记者团体和独立调查机构依据“信息自由法案”向国会多次请愿才得以公开的，因此具有极高的社会关注度与新闻时效性。但是，这些档案是不可检索的低质量PDF文件，若想将其转换为可以检索并便于浏览的文件格式，需要进行再处理。华盛顿邮报希望将这些档案在第一时间上传到互联网，以便公众查询，但是据估算仅每一页的操作，以报社现有的计算能力就需要30分钟。因此，华盛顿邮报将这个档案的转换工程交给Amazon EC2（Elastic Compute Cloud）。Amazon EC2同时使用200个虚拟服务器实例，每个服务器的单页平均处理时间都缩短为一分钟，并在9小时内将所有的档案转换完毕，以最快的速度将这些第一手资料呈现给读者。

[案例二] Giftag是一款Web 2.0应用，它能被以插件的形式安装在Firefox和IE浏览器上。互联网用户在浏览网页，尤其是在浏览购物网站的时候，可以利用这个插件将心仪的商品加入到由Giftag维护的商品清单中，并将这个清单与好友分享。这个应用一经推出，便广泛流行起来，注册用户数量激增，每天Giftag的服务器都要响应数以百万计的请求，并存储用户提交的海量信息，没过多久服务器就不堪重负。后来，Giftag将应用迁移到Google App Engine（GAE）平台，基于GAE的开放API，Giftag可以利用Google具有可伸缩性的计算处理性能来响应高峰期的用户请求，利用Google的分布式数据库来存储用户数据，甚至可以使用Gmail邮箱和Google的搜索功能来增强用户体验。Giftag实现了从一个初创的Web 2.0应用向一个稳定的、持续增长的网络服务的平稳过渡。

[案例三] 哈根达斯是著名的冰激凌供应商，其加盟店遍布世界各地。

因此，公司需要一个CRM（客户关系管理）系统对所有的加盟店进行管理。当时哈根达斯用Excel表单来管理和跟踪主要的加盟店，用Access数据库来存储协议加盟店的数据，但是使用虚拟专用网（VPN）来访问该数据库的效果总是不太好。因此，公司急需一个能够让分布在各地的员工沟通协作的解决方案，并且该方案应该能够根据不同的需求进行灵活配置。哈根达斯公司选择了Salesforce CRM企业版，应用系统在不到6个月的时间就上线了。除此之外，该系统将Microsoft Outlook和Salesforce CRM集成了起来，从而使员工能够轻松地访问Outlook中的联系人列表、日程和商业信息。Salesforce.com还为哈根达斯的解决方案提供了员工培训模块、加盟店的跟踪模块，以及新店选址模块。哈根达斯公司用更少的成本获得了超预期的效果。





[案例四] 国际商业机器公司（IBM）作为全球整合的大型跨国企业，在全球共拥有8所研究院，汇聚了3000多位顶尖级的科学家和研究员。在他们之中共有6位诺贝尔奖获得者和6位图灵奖获得者，其中仅2008年一年就有4186项专利从这8所研究院里诞生。在这里，每天都有不计其数的科学实验在进行着，其中有些实验需要有海量的计算和存储资源作为支撑。虽然每所研究院都配备了先进的IT设备，但仍然满足不了某些实验的需求。除此之外，由于这些研究院分布在世界各地，处于不同的时区和大陆，为合作科研提出了挑战。为了给研究部门的创新提供源源不断的支持，也为提高各研究院间的沟通协作效率，IBM公司构建了IBM Research Compute Cloud（RC2）将分散在各个研究院的资源系统（如服务器、存储）整合，为公司内部所使用。该系统为科研人员提供了共享计算和存储资源的平台，通过任务调度和安排，为每一项科学实验提供了有保障的动态资源供给，而且不需要科学实验人员来管理这些资源。不仅如此，不论是实验的中间流程还是最终结果都是在该系统中完成和保存的，所以有效地保证了数据的安全，并使得身处世界各地的研究人员随时可以对它们进行查询和交换。这一切大大提高了协同科研的效率，为IBM公司不断深入的创新提供了强大的推动力。

## 2. 案例分析

在案例一中，如果没有Amazon EC2提供的计算能力，华盛顿邮报社按照其所拥有的资源，需要超过一年的时间来完成全部档案的格式转

换工作。显然，这样的效率不能满足新闻的时效性和公众对于信息的期盼。恰恰是Amazon公司通过其EC2平台，将计算资源打包提供给客户，使报社可以在9小时内就得到了1407小时的虚拟服务器机时，在第一时间完成了档案的转换，而华盛顿邮报社仅需要向Amazon公司支付144.62美元的费用。

在案例二中，Giftag公司和其他初创型Web 2.0公司一样，面临着高昂的基础设施投入费用，如购置服务器、租用带宽等。而基础设施的投入往往是不易估量的，如果一次投入过大而应用并没有达到预期的流行度，就会造成投资的浪费；反之，如果应用获得了超预期的反响，用户数量激增，那么就会给服务器、带宽带来巨大的压力，从而造成应用服务质量下降和客户的流失。而且Web应用需要复杂的软件配置，包括数据库、中间件、Web服务器等要素，如果其中一项配置得不合理，就会产生连锁反应，影响整个应用的表现。这些潜在问题都给创业公司提出了巨大的挑战。在GAE平台上，Giftag可以将自己的精力集中于应用本身，而将诸如服务器动态扩展、数据库访问、负载均衡等各个层次的问题交给GAE平台来解决。正是由于GAE将Web应用所需的基础功能作为服务提供给了Giftag，才使得其可以专注于应用的开发和优化。

在案例三中，哈根达斯公司要搭建自己的CRM平台，传统的做法是先聘请一支专业的顾问团队研究公司的业务流程，建模分析并提出咨询报告。然后再雇用一家IT外包公司，进驻自己的公司对平台进行开发，其间那些需求→设计→实施→需求变更→再设计→再实施的循环可能会多次出现。同时，哈根达斯作为一家冰淇淋制作厂商，还需要投资IT设备，如购买服务器、交换机、防火墙、各种各样的软件，以及租用带宽等，为系统上线做准备。最后，经历了这令人精疲力竭的过程后系统终于上线了，但它是不是真的满足了哈根达斯公司最初的愿望呢，可能永远不会有人知道和提起了。幸运的是，哈根达斯公司没有重复这条被别的公司走过无数次的老路。Salesforce.com作为CRM系统的专业提供商，对这个领域有着精深的理解。同时，它能够将已经完成的CRM应用模块打包，供用户选择。用户只需要如同在超市选购商品一样选择自己需要的功能模块，让Salesforce.com进行定制集成，一个属于自己的CRM系统就完成了，系统的上线和维护也将由Salesforce.com的专业团队负责。这样，一家非IT公司就可以专注于它的主营业务，使IT真正成为公司的支撑，而不是拖累。

在案例四中，IBM公司分布在世界各地的8所研究院虽然各自拥有强大的IT基础设施，但有时单个科学实验对资源的需求超出了其所在研究院所具有的资源规模，而且以往各自分割独立的组织方式很难让各个机构间协作完成一项工作。实际上，蓝色巨人IBM一直在努力整合自己的IT资源，以降低运营成本。早在2007年，IBM公司就开始着手将运行在3900台服务器上的业务迁移到30台大型机上，从而减少了80%的电力消耗，同时也促进了公司业务的整合。IBM Research Compute Cloud（RC2）的建立把分散于各地的资源从物理和逻辑上整合在一起，为研究院的科研提供了一个近乎取之不尽的资源池。此外，计算资源的整合带动了业务的整合，研究员们可以在IBM RC2上共享实验所需的工具、平台甚至是结果，大大加速了科研的进程。值得注意的是，与前三个案例不同，IBM RC2是供IBM公司内部使用的私有系统，而不是一个为公司以外的用户提供服务的第三方公用平台。

通过以上四个典型案例，相信读者已经初步领略到了云计算的魅力和价值。是的，云计算就是一种更加智慧的信息技术，它化繁为简、化难为易、化不可能为可能。下面，我们将给出“云计算”的定义。

## 5.1.2 云计算的定义

### 1. 云计算的来源

在云计算最早被提出的时候，曾经有一种流行的说法来解释“云计算”为何被称为“云”计算：在互联网技术刚刚兴起的时候，人们画图时习惯用一朵云来表示互联网，因此在选择一个名词来表示这种基于互联网的新一代计算方式的时候就选择了“云计算”这个名词。虽然这个解释非常有趣和浪漫，但是却容易让人们陷入云里雾中，不得其正解。

进入互联网时代后，人们热衷于上网冲浪，通过浏览网页来获得资讯。当用户在浏览器上输入网址后，浏览器将会与DNS服务器和网站服务器进行一系列的交互，将网页内容呈现在用户面前，而这些交互过程是通过互联网经过多次路由转发最终完成的。因为这个过程对用户是透明的，所以当时人们在绘制互联网示意图时，将网络抽象成一朵云，意在不去关心网络的转发过程，而去关注服务器端和客户端，如图5.1所示。

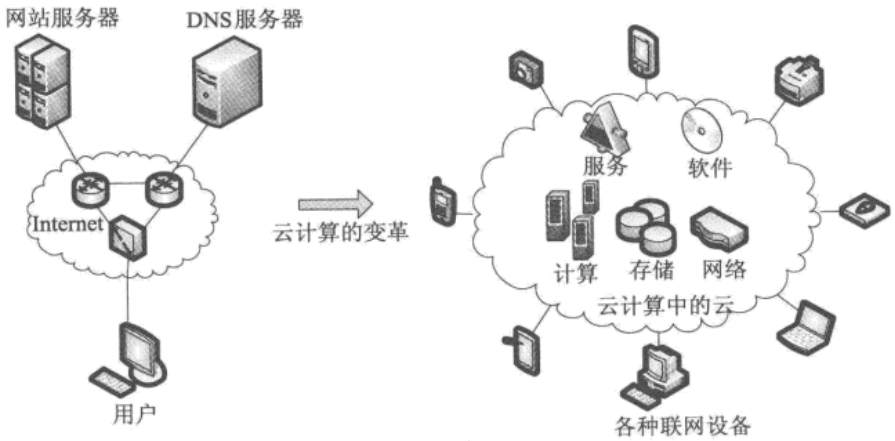



图5.1　云计算中的“云”

随着互联网的发展，带宽得到了显著提高，无线接入方式也变得丰富起来，除了个人电脑外，越来越多的设备已经具有了接入互联网的能力，比如移动电话、办公设备甚至是家用电器。同样，互联网的作用也不再局限于浏览网页和收发电子邮件，还能够为企业提供诸如电子商务、客户关系管理等信息服务；为普通用户提供诸如博客、视频等服务；为科研机构提供强大的计算处理功能。因此，互联网的含义变得充实起来，除了人们普遍认知的接入、路由等含义，还包括了计算、存储、服务和软件等元素。因此，“云计算”这个名词就应运而生了。从图5.1中我们可以看出，云计算中的“云”不仅包含了网络，更包含了那些曾经被描绘在云外的事物。这个小小的改变在图上看似简单，实际上蕴含着深刻的变革。

正如用云描绘网络来强调对网络的运用而非关注于其实现细节一样，云计算用云描绘包括网络、计算、存储等在内的信息服务基础设施，以及包括操作系统、应用平台、Web服务等在内的软件，就是为了强调对这些资源的运用，而不是它们的实现细节。

## 2. 什么是云计算

了解了云计算为什么被称之为“云”之后，下面我们将给出云计算的定义。其实，这个概念被提出的时间并不长，然而对这个概念的定义却是百家争鸣。这体现了云计算包罗万象的特质，也说明业界对它的重视——既然所有人都希望成为云计算产业链中的一个角色，自然都会从自身的角度出发来定义云计算，那么对于概念的提取就是一个求同存异的过程。下面，我们先列举一些为人们普遍认可的云计算定义，然后再给出本书的定义。

维基百科（Wikipedia.com）认为云计算是一种能够将动态伸缩的虚拟

化资源通过互联网以服务的方式提供给用户的计算模式，用户不需要知道如何管理那些支持云计算的基础设施。

Whatis.com认为云计算是一种通过网络连接来获取软件和服务的计算模式，云计算使得用户可以获得使用超级计算机的体验，用户通过笔记本电脑与手机上的瘦客户端接入云中获取需要的资源。

美国加州大学伯克利分校最近发表了一篇关于云计算的报告，该报告认为云计算既指在互联网上以服务形式提供的应用，也指在数据中心中提供这些服务的硬件和软件，而这些数据中心中的硬件和软件则被称为云。

商业周刊（BussinessWeek.com）发表文章指出，Google的云就是由网络连接起来的几十万甚至上百万台的廉价计算机，这些大规模的计算机集群每天都处理着来自于互联网上的海量检索数据和搜索业务请求。商业周刊在另一篇文章中总结说，从Amazon的角度看，云计算就是在一个大规模的系统环境中，不同的系统之间相互提供服务，软件都是以服务的方式运行，当所有这些系统相互协作，并在互联网上提供服务时，这些系统的总体就成为了云。

Salesforce.com认为云计算是一种更友好的业务运行模式。在这种模式中，用户的应用程序运行在共享的数据中心中，用户只需要通过登录和个性化定制就可以使用这些数据中心的应用程序。

IBM认为云计算是一种共享的网络交付信息服务的模式，云服务的使用者看到的只有服务本身，而不用关心相关基础设施的具体实现。本书沿用IBM的定义，云计算是一种革新的IT运用模式。这种运用模式的主体是所有连接着互联网的实体，可以是人、设备和程序。这种运用方式的客体就是IT本身，包括我们现在接触到的，以及会在不远的将来出现的各种信息服务。而这种运用方式的核心原则是：

硬件和软件都是资源并被封装为服务，用户可以通过互联网按需地访问和使用。

在云计算中，IT业务通常运行在远程的分布式系统上，而不是在本地计算机或者单个服务器上。这个分布式系统由互联网相互连接，通过开放的技术和标准把硬件和软件抽象为动态可扩展、可配置的资源，并对外以服务的形式提供给用户。该系统允许用户通过互联网访问这些服务，并获取资源。服务接口将资源在逻辑上以整合实体的形式呈现，隐蔽其中的实现细节。该系统中业务的创建、发布、执行和管理都可以在网络上进行，

而用户只需要按资源的使用量或者业务规模付费。

### 3. 云计算的特征

在云计算的定义中，有四个关键要素，如图5.2所示。

第一点，硬件和软件都是资源，通过互联网以服务的方式提供给用户。正如上一小节所描述的，Amazon EC2将计算处理能力打包为资源提供给用户；Google App Engine将从设计开发到部署实施Web应用所需的软件、硬件平台一起打包提供给用户；Salesforce.com CRM 将专业的客户关系管理应用模块打包为解决方案提供给用户。在云计算中，资源已经不限定在诸如处理器机时、网络带宽等物理范畴，而是扩展到了软件平台、Web服务和应用程序的软件范畴。传统模式下自给自足的IT运用模式，在云计算中已经改变成为分工专业、协同配合的运用模式。对于企业和机构而言，他们不再需要规划属于自己的数据中心，也不需要将精力耗费在与自己主营业务无关的IT管理上。相反，他们可以将这些功能放到云中，由专业公司为他们提供不同程度、不同类型的信息服务。对于个人用户而言，也不再需要一次性投入大量费用购买软件，因为云中的服务已提供了他所需要的功能。

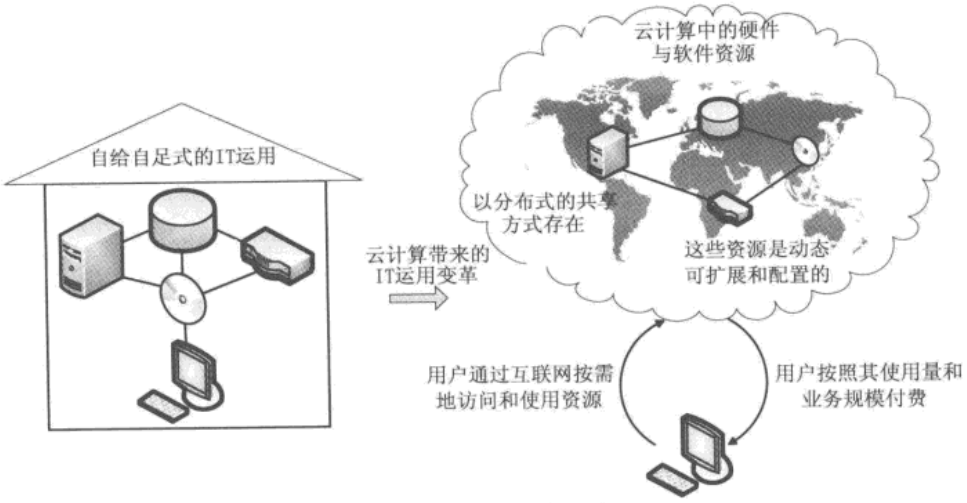



图5.2　云计算的特征

第二点，这些资源都可以根据需要进行动态扩展和配置。例如在上一小节的典型案例中，Amazon EC2可以在极短的时间内为华盛顿邮报社初始化200台虚拟服务器的资源，并在9小时的任务完成后快速地回收这些资源；Google App Engine可以满足Giftag的快速增长，不断为其提供更多的存储空间、更高的带宽和更快速的处理能力；Salesforce.com可以为哈根达斯公司在已经成型的CRM系统中动态地添加和删除应用模块，来满足客户不

断改进的业务需求。这些例子都体现了云计算可动态扩展和配置的特性。

第三点，这些资源在物理上以分布式的共享方式存在，但最终在逻辑上以单一整体的形式呈现。对于分布式的理解有两个方面。一方面，计算密集型的应用需要并行计算来提高运算效率。例如，一个Web应用是由多个服务器通过集群的方式来实现的，此类的分布式系统，往往是在同一个数据中心中实现的，虽然有较大的规模，由几千甚至上万台计算机组成，但是在地域上仍然相对集中。另一方面，就是地域上的分布式。例如，一款商业应用的服务器可以设在位于纽约的华尔街，但是它的数据备份却由位于德州戈壁中的数据中心完成。在上文的典型案例四中，IBM公司在世界范围内共拥有8所研究院，IBM RC2将这些研究院中的数据中心通过企业内部网连接起来，为世界各地的研究员提供服务。作为最终用户，这些研究员们并不知道也不关心某一次科学运算运行在哪个研究院的哪台服务器上，因为云计算中分布式的资源向用户隐藏了实现细节，并最终以单一整体的形式呈现给用户。

最后，用户按需使用云中的资源，按实际使用量付费，而不需要管理它们。例如，在上一小节的例子中，华盛顿邮报社为尽快完成档案的转换任务，使用了200台虚拟服务器，并为其所获得的1407小时机时支付了144.62美元。虽然华盛顿邮报社没有足够的运算处理能力，但是云给了它强大的资源以帮助其快速完成任务，而它仅需要根据实际使用量来付费。对于华盛顿邮报社来说，如此巨大计算量的任务并不经常出现，因此按照这个标准购置IT设备显然是不合理的。如果没有Amazon EC2，华盛顿邮报社在 9 小时内完成档案的转换工作将是不可能完成的任务。同样，在Giftag的例子中，Giftag需要做的仅仅是根据其业务的增长而使用更多的Google App Engine的资源。依托Google强大的数据中心，Giftag拥有近乎无限的资源来满足新用户的注册，从而避免了自己投资IT基础设施而可能出现的浪费现象或客户流失。

总之，在云计算中软、硬件资源以分布式共享的形式存在，可以被动态地扩展和配置，最终以服务的形式提供给用户。用户按需使用云中的资源，不需要管理，只需按实际使用量付费。这些特征决定了云计算区别于自给自足的传统IT运用模式，必将引领信息产业发展的新浪潮。

### 5.1.3 云计算的分类

以上我们分析了云计算中“云”的含义，给出了云计算的例子和定

义，并总结了云计算的关键特征。在云计算中，硬件和软件都被抽象为资源并被封装为服务，向云外提供。用户以互联网为主要接入方式，获取云中提供的服务。细心的读者可能已经发现，本章开始给出的四个案例之间既有共同点又存在着差别。相同点是，它们都获取了云中的服务，快速、高效地完成了工作；不同点是，它们获取的服务类型不尽相同。下面我们分别从云计算提供的服务类型和服务方式的角度出发，为云计算分类。

### 1. 按服务类型分类

所谓云计算的服务类型，就是指其为用户提供什么样的服务；通过这样的服务，用户可以获得什么样的资源；以及用户该如何去使用这样的服务。目前业界普遍认为，以服务类型为指标，云计算可以分为以下三类，如图5.3所示。


图5.3　云计算的服务类型

- **基础设施云（Infrastructure Cloud）**。例如在上文案例一中提到的Amazon EC2。这种云为用户提供的是底层的、接近于直接操作硬件资源的服务接口。通过调用这些接口，用户可以直接获得计算和存储能力，而且非常自由灵活，几乎不受逻辑上的限制。但是，用户需要进行大量的工作来设计和实现自己的应用，因为基础设施云除了为用户提供计算和存储等基础功能外，不进一步做任何应用类型的假设。
- **平台云（Platform Cloud）**。例如在上文案例二中提到的Google App Engine。这种云为用户提供一个托管平台，用户可以将他们所开发和运营的应用托管到云平台中。但是，这个应用的开发和部署必须遵守该平台特定的规则和限制，如语言、编程框架、数

据存储模型等。通常，能够在该平台上运行的应用类型也会受到一定的限制，比如Google App Engine主要为Web应用提供运行环境。但是，一旦客户的应用被开发和部署完成，所涉及的其他管理工作，如动态资源调整等，都将由该平台层负责。

- 应用云（Application Cloud）。例如在上文案例三中提到的Salesforce.com。这种云为用户提供可以为其直接所用的应用，这些应用一般是基于浏览器的，针对某一项特定的功能。应用云最容易被用户使用，因为它们都是开发完成的软件，只需要进行一些定制就可以交付。但是，它们也是灵活性最低的，因为一种应用云只针对一种特定的功能，无法提供其他功能的应用。

表5.1总结了从服务类型的角度来划分的云计算类型。实际上，正如我们现在所熟悉的软件架构范式，自底向上依次为计算机硬件—操作系统—中间件—应用一样，这种云计算的分类也暗含了相似的层次关系。这里不同类型的云其实就是云的不同层次提供的云计算服务，我们将在第6章从技术的角度详细分析云计算的层次架构，给出每一层次的主要功能和实现示例。

表5.1 按服务类型划分云计算

| 分　类 | 服务类型 | 运用的灵活性 | 运用的难易程度 |
|---|---|---|---|
| 基础设施云 | 接近原始的计算存储能力 | 高 | 难 |
| 平台云 | 应用的托管环境 | 中 | 中 |
| 应用云 | 特定功能的应用 | 低 | 易 |

## 2. 按服务方式分类

云计算作为一种革新性的计算模式，虽然具有许多现有模式所不具备的优势（云计算带来的优势将在下文具体分析），但是也不可否认地带来了一系列挑战，不论是从商业模式上还是从技术上。首先就是安全问题，对于那些对数据安全要求很高的企业（如银行、保险、贸易、军事等）来说，客户信息是最宝贵的财富，一旦被人窃取或损坏，后果将不堪设想。其次就是可靠性的问题，例如银行希望其每一笔交易都能快速、准确地完成，因为准确的数据记录和可靠的信息传输是让用户满意的必要条件。还有就是监管问题，有的企业希望自己的IT部门完全被公司所掌握，不受外界的干扰和控制。虽然云计算可以通过系统隔离和安全保护措施为用户提供有保障的数据安全，通过服务质量管理来为用户提供可靠的服务，但是仍有可能不能满足用户的所有需求。

针对这一系列问题，业界按照云计算提供者与使用者的所属关系为划分标准，将云计算分为三类，即公有云、私有云和混合云，如图5.4所示。用户可以根据其需求，选择适合自己的云计算模式。


图5.4　云计算的服务方式

- **公有云。**公有云是由若干企业和用户共享使用的云环境。如我们上文所举的案例一、案例二和案例三都属于公有云的范畴。在公有云中，用户所需的服务由一个独立的、第三方云提供商提供。该云提供商也同时为其他用户服务，这些用户共享这个云提供商所拥有的资源。
- **私有云。**私有云是由某个企业独立构建和使用的云环境。如我们上文所举的案例四。私有云是指为企业或组织所专有的云计算环境。在私有云中，用户是这个企业或组织的内部成员，这些成员共享着该云计算环境所提供的所有资源，公司或组织以外的用户无法访问这个云计算环境提供的服务。
- **混合云。**指公有云与私有云的混合。

一般来说，对安全性、可靠性及IT可监控性要求高的公司或组织，如金融机构、政府机关、大型企业等，是私有云的潜在使用者。因为他们已经拥有了规模庞大的IT基础设施，因此只需进行少量的投资，将自己的IT系统升级，就可以拥有云计算带来的灵活与高效，同时有效地避免使用公有云可能带来的负面影响。除此之外，他们也可以选择混合云，将一些对安全性和可靠性需求相对较低的应用，如人力资源管理等，部署在公有云上，来减轻对自身IT基础设施的负担。相关分析指出，一般中小型企业和创业公司将选择公有云，而金融机构、政府机关和大型企业则更倾向于选择私有云或混合云。

## 5.1.4 相关概念辨析

在计算机科学技术发展的历史上，曾经出现过一些里程碑式的技术。

这些技术产生的时间或远或近，但都对当今世界的IT运用模式产生了巨大的影响。这些技术包括并行计算、网格计算和效用计算。罗马不是一天建成的，同样，云计算也不是一蹴而就的，而是从这些技术中逐渐演进而来，既一脉相承，又有所不同。下面我们就来辨析云计算与这些相关概念的异同。

### 1. 并行计算

并行计算（Parallel Computing）将一个科学计算问题分解为多个小的计算任务，并将这些小任务在并行计算机上同时执行，利用并行处理的方式达到快速解决复杂运算问题的目的。并行计算一般应用于诸如军事、能源勘探、生物、医疗等对计算性能要求极高的领域，因此也被称为高性能计算（High Performance Computing）或超级计算（Super Computing）。并行计算机是一群同构处理单元的集合，这些处理单元通过通信和协作来更快地解决大规模计算问题。常见的并行计算机系统结构包括共享存储的对称多处理器（SMP）、分布式存储的大规模并行机（MPP）和松散耦合的分布式工作站集群（COW）等。解决计算问题的并行程序往往需要特殊的算法，编写并行程序需要考虑很多问题之外的因素，例如各个并发执行的进程之间如何协调运行、任务如何分配到各个进程上运行等。

并行计算机可以说是云环境的重要组成部分，例如案例四中IBM研究院科研人员使用的IBM RC2。与云计算的思想相似，目前世界各国已经集中建立了若干超级计算中心来服务于该区域内有并行计算需求的用户，并采用分担成本的方式进行付费。但是，云计算与传统意义上的并行计算相比，又存在明显的区别。首先，并行计算需要采用特定的编程范例来执行单个大型计算任务或者运行某些特定应用，而云计算需要考虑的是如何为数以千万计的不同种类应用提供高质量的服务环境，以及如何提高这个环境对用户需求的响应从而加速业务创新。一般来说，云计算对用户的编程模型和应用类型等没有特殊限定，用户不再需要开发复杂的程序，就可以把他们的各类企业和个人应用迁移到云计算环境中。其次，云计算更加强调用户通过互联网使用云服务，而在云中利用虚拟化进行大规模的系统资源抽象和管理。在并行计算中，计算资源往往集中在单个数据中心的若干台机器或者是集群上。云计算中资源的分布更加广泛，正如上文所述，它已经不再局限于某个数据中心，而是扩展到了多个不同的地理位置。同时，由于采用了虚拟化技术，云计算中的资源利用率可以得到有效的提

升。由此可见，云计算是互联网技术和信息产业蓬勃发展背景下的产物，完成了从传统的、面向任务的单一计算模式向现代的、面向服务的多元计算模式的转变。

### 2. 网格计算

网格计算（Grid Computing）是一种分布式计算模式。网格计算技术将分散在网络中的空闲服务器、存储系统和网络连接在一起，形成一个整合系统，为用户提供功能强大的计算及存储能力来处理特定的任务。对于使用网格的最终用户或应用程序来说，网格看起来就像是一个拥有超强能力的虚拟计算机。网格计算的本质在于以高效的方式来管理各种加入了该分布式系统的异构松耦合资源，并通过任务调度来协调这些资源合作完成一项特定的计算任务。

可见，网格计算着重于管理通过网络连接起来的异构资源，并保证这些资源能够充分为计算任务服务。通常，用户需要基于某个网格的框架来构建自己的网格系统，并对其进行管理，在其上执行计算任务。云计算则不同，用户只需要使用云中的资源，而不需要关注系统资源的管理和整合。这一切都将由云提供者进行处理，用户看到的是一个逻辑上单一的整体。因此，在资源的所属关系上存在着较大差异，也可以说在网格计算中是多个零散资源为单个任务提供运行环境，而在云计算中是单个整合资源为多个用户提供服务。

### 3. 效用计算

效用计算（Utility Computing）强调的是IT资源，如计算和存储等，能够根据用户的要求被按需地提供，而且用户只需要按照其实际使用情况付费。效用计算的目标是IT资源能够像传统公共设施（如水和电等）一样的供应和收费。效用计算使得企业和个人不再需要一次性的巨额投入就可以拥有计算资源，而且能够降低使用和管理这些资源的成本。效用计算追求的是提高资源的有效利用率，最大程度地降低资源的使用成本和提高资源使用的灵活性。

效用计算所提倡的资源按需供应、用户按使用量付费的理念与云计算中的资源使用理念相符。云计算也可以按照用户的资源需求分配运算、存储、网络等各种基础资源。比效用计算更进一步的是，云计算已经有了很多实际应用案例，所涉及的技术和架构可行性更强。云计算所关注的是如

何在互联网时代以其自身为平台开发、运行和管理不同的服务。云计算不但注重基础资源的提供，而且注重服务的提供。在云计算环境中，不但硬件等IT基础资源能够以服务的形式来提供，应用的开发、运行和管理也是以服务的形式提供的，应用本身也可以采用服务的形式来提供。因此，云计算与效用计算相比，技术和理念所涵盖的范围更广泛、可行。

## 5.2 云计算的优势与带来的变革

正如达尔文在其著作《物种起源》中指出的那样，自然界中的生物是按照物竞天择、适者生存的规律一步一步进化而来的，优秀的物种会发展出适应当前环境的特征，而正是这些特征使得该物种能够战胜残酷的生存考验，最终繁衍下来。云计算作为互联网时代最新提出的IT运用模式，必然要顺应历史潮流，体现技术进步，才能在IT这个高速发展的产业里成长起来。我们将在本节分析云计算的优势与带来的变革，相信读者阅读过本节后，将对云计算的未来充满信心。

### 5.2.1 云计算的优势

本节将按照从商业到技术的顺序，首先在IT产业的层面，从优化产业布局和推进专业分工的角度分析云计算的优势，再逐渐深入到云计算的运行和维护层面，从提升资源效率、减少运营投资、降低管理成本的角度分析云计算的优势。

#### 1. 优化产业布局

正如上文所述，云计算将企业原先自给自足的IT运用模式改变为由云计算提供商按需供给的模式。IT业界将出现一些实力雄厚的云计算提供商，他们拥有雄厚的技术实力和管理经验，雇用专业的商业专家和研发人员。最重要的是，他们有一座甚至许多座规模巨大的计算中心来支撑云中的服务。在摩尔定律的指引下，硬件成本正在不断降低，这些未来的云计算运营商心目中的关键资源不再是服务器，而是运行这些服务器所必不可少的电力资源。

以正在大规模投资云计算的Google公司为例，据推测（关于数据中心的建造细节一直以来被各个公司列为商业机密）该公司在多个地点拥有约12座专属数据中心，在这些数据中心中一共同时运行着约100万台服务器。

每一台服务器加上为其提供冷却的空调的功率是500瓦特。就算忽略包括路由器、交换机在内的其他设备的电力消耗，该公司数据中心每小时需要使用的电力也足有500兆瓦，这相当于半个旧金山市区的电力消耗。

电力作为一种传统资源，分布很不均匀。由于自然禀赋和政策法规的影响，各地的电价具有很大差异，如表5.2所示。在当前技术条件下，用电网传送电力的成本和产生的浪费要远大于用互联网传输数据，而电价又忠实地反映了获取电力资源的难易程度，因此云计算提供商在建立大规模数据中心的时候都会充分考虑这个因素，将大型数据中心建造在电力资源丰富、地理条件安全、很少有自然灾害的地方；同时又要充分考虑诸如当地法律政策、是否靠近互联网重要结点等非自然因素。例如据报道称，有几座大型数据中心正在爱荷华州的Council Bluffs与华盛顿州的Quincy建设，而这些地点的选择恰好体现了对于以上因素的充分考虑。

表5.2 美国部分地区电价比较

| 电价（美分/千瓦时） | 地　　点 | 可能的原因 |
| --- | --- | --- |
| 3.6 | 爱达荷州 | 丰富的水电资源 |
| 10.0 | 加州 | 电网远距离传输，州法律禁止火电厂的建造 |
| 18.0 | 夏威夷 | 油料需要通过海运进岛进行发电 |

可见，进入云计算时代后，IT已经从以前那种自给自足的作坊模式，转化为具有规模化效应的工业化运营，一些小规模的单个公司专有的数据中心将被淘汰，取而代之的是规模巨大而且充分考虑资源合理配置的大规模数据中心。而正是这种更迭，生动地体现了IT产业的一次升级，从以前分散的、高耗能的模式转变为集中的、资源友好的模式，顺应了历史发展的潮流。

### 2. 推进专业分工

正如上一小节所述，不同于中小型企业的数据中心只能在距离企业不太远的地方选址以便维护，专业公司的大型数据中心可以充分利用选址灵活的优势合理配置资源。此外，大型数据中心具有实力雄厚的科研技术团队、丰富的维护管理经验来体现专业分工的优势。

云计算提供商普遍采用大规模数据中心，比中小型数据中心更专业，管理水平更高，提供单位计算所需的成本更低廉，如表5.3所示。中小规模

的数据中心采用风冷的方式进行温度调节，空调耗电量较大，而大型数据中心一般采用专业的水冷与风冷相结合的方式进行温度调节，这样的数据中心一般建立在水资源丰富的河边，将用于制冷的水抽取到制冷单元，当水温度升高后再送到室外自然冷却，相对风冷来说这是一种既节能环保又经济的温度调节方式。

表5.3 大型数据中心与中小型数据中心相比的成本优势

| 数据中心属性 | 中小型数据中心 | 大型数据中心 |
| --- | --- | --- |
| 服务器个数 | < 2000 | > 2000 |
| 每个管理员管理服务器数 | < 500 | > 500 |
| PUE值 | 2.0～2.5 | 1.0～1.5 |
| 服务器供电方式 | 交流电 | 直流电 |
| 电价 | 高 | 低 |
| 制冷方式 | 风冷 | 水冷 + 风冷 |
| 提供单位计算力的成本 | 高 | 低 |

同时，专业的云计算提供商可以有更多的科研和经费投入来推动数据中心的技术革新。例如，目前大多中小型数据中心采用交流电的供电方式，仅能达到约75%的能效比，其中有25%的电能被白白浪费，转化成了热量，加剧了温度调节所需的能源消耗。但通过技术革新，改用直流电源的方式进行供电，仅此一项，大型专业数据中心就可以节电约30%。

除了在硬件上更加专业，云计算提供商还具有更完善的软件，这包括具有丰富知识和经验的管理团队及与其配套的管理软件。在中小型的数据中心，平均每个工作人员最多可以管理170台服务器。而在大型数据中心中，由于有专业团队和工具的支持，每个工作人员可以同时管理的服务器数量达1000台以上。因此，人力成本这一项可以被大幅度削减。

由此可见，云计算带来的是更加专业的分工，更进一步优化的IT产业格局。通过让专业的人做专业的事，各取所长，扬长避短，有效避免了IT产业中可能产生的内耗。另一方面，专业分工也孕育了新的产业契机，除了现有的大型IT公司外，一批新兴的高科技企业也将在云计算中找到自己的位置并逐渐成长起来。

### 3. 提升资源利用率

前面我们在IT产业的层面，从产业布局和专业分工的角度阐述了云计

算的优势。下面我们将深入到云计算所涉及的各个实体，讨论这个新兴的计算模式将赋予它们怎样的优势。

在云计算模式下，高科技企业、传统行业甚至是互联网公司的IT业务都可以在不同程度上外包给专业的云计算提供商进行管理。如在上文介绍的典型案例二中，Giftag公司就将其设计的Web 2.0应用交由Google App Engine托管，Google公司根据其业务量的变化来调整和分配其所需的资源。值得注意的是，Giftag并不是Google App Engine平台上唯一的托管应用。实际上，它与成千上万其他的Web应用一起共享这个平台提供的服务与资源。

图5.5是一个Web应用的典型负载变化图，从图中可以看出负载呈现出三个主要规律：其一是负载的周期性变化规律，通常由昼夜差异和周末与工作日的差异引起，基本可以通过长期观察来预测；其二是一次性任务或突发事件引起的负载，例如某热门话题会引起网站的访问量激增，通常无法预测；其三是由于业务增长引起的负载长期增长趋势，一定程度上可以预测。面对这样变化的负载，传统的Web应用提供商或者企业专有数据中心应该如何来规划资源呢？一般来说无外乎图中的A、B、C三种方式。方式A仅考虑短期的负载来分配资源，该方式产生的浪费最少，仅在负载周期低谷时有较大资源浪费。然而，在不对业务发展进行预测的情况下分配资源，会导致一段时间后因资源不足影响业务系统运行，或不久就需要再次扩容，带来管理上的复杂性。目前被采用较多的是方式B，这种方式考虑了负载长期的增长趋势，有一定预见性地增加了资源，但相比方式A来说，造成短期内一定的资源浪费。方式C为了应对不可预测的突发事件或一次性任务而准备大量资源，在绝大多数情况下资源处于严重浪费的状态。这种方式仅适用于业务系统极其重要、为保证可用性可以不计成本的应用。

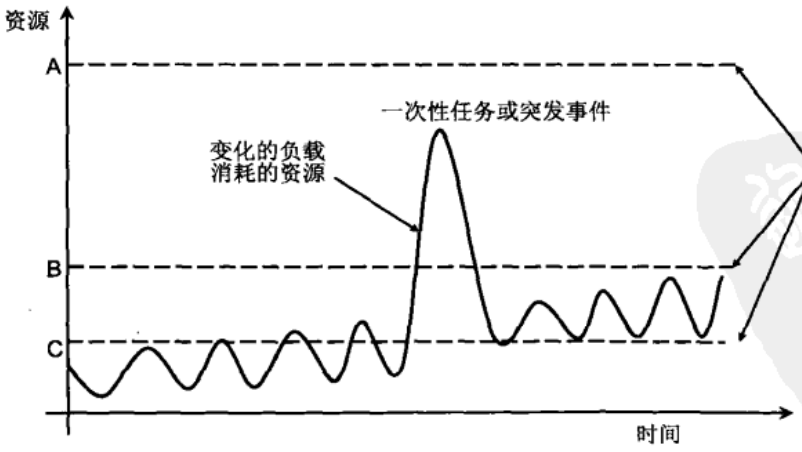



图5.5 典型的业务系统负载变化及传统的资源分配方式

可见，传统的数据中心无法兼顾业务的可用性和资源的高效利用，只能在二者之间达到某种程度的平衡。一般来说，企业为了保证业务系统的高可用性，会牺牲掉资源的高效性。据统计，多数企业数据中心的资源利用率在15%以下，有的还不到5%。而在云计算的平台中，若干企业的业务系统共用同一个大的资源池，资源池的大小可以适时调整，还可以通过动态资源调度机制对资源进行实时的合理分配。即使有突发事件对某一个业务系统产生冲击，也不会对整个资源池造成很大影响。通过这些手段，云计算平台中的资源利用率可达80%以上，与传统数据中心的资源利用率相比有大幅度提升。

### 4. 减少初期投资

从云服务提供商的角度看，同时托管多个服务提高了资源利用率，也降低了其长期的运营成本。同样，对于将自己的IT业务外包给云计算提供商的公司，他们的一次性IT投入也降到了最低，从而有效地规避了财务风险。

云计算将取代传统的企业专有数据中心。企业无需拥有硬件，而是直接使用云中的计算资源。云计算即用即付费的方式消除了企业的一次性投入，包括数据中心的营建，以及硬件设备的购置和定期更换。这种一次性投入对企业的现金流冲击较大，它意味着企业预付了若干年的投入。IT设备的平均寿命是3～5年；制冷设备、监控设备、门禁系统等其他设备的使用寿命则是10～20年；如果再考虑上数据中心的建筑寿命，就可以达到几十年之久。这样巨额的一次性投入将使企业背负沉重的负担。此外，一旦企业发生较大变化，如业务转型、系统下线、政策变化等，前期投入的资产就有可能面临被折价处置的困境。

在大多数情况下，软件同样也是一项高昂的支出。如果需要一套高质量的行业解决方案，企业首先要购买构建该解决方案所必须的中间件软件的许可证，然后在这个基础上购买或者开发自己所需要的特定解决方案。除此之外，当这些服务器或者软件被购入以后，很多时候它们其实并没有被充分利用。因为系统的负载是不均衡的，甚至有些时候系统是空闲的，即并不处理任何用户请求。

回顾本章开始时华盛顿邮报的例子，显然以报社现有的IT资源是无法完成档案格式转换工作的。但是，报社也不可能为了这个任务而进行一次性投入，购买功能强大的计算机设备。而恰好是云计算提供的“按使用付费”的计价模型有效地降低了用户的IT成本，使不可能的任务成为可能。

云计算帮助用户降低IT成本体现在两个方面：第一，用户不再需要进行巨大的一次性IT投资，彻底省去了购置、安装、管理软、硬件的费用，因为他们可以从云计算提供商那里租用这些IT基础设施；第二，用户在使用这些IT资源时，可以按照自己的实际使用量付费。如表5.4所示列出了Amazon 公司提供的打包计算资源实例及他们的计价标准。在这种计价模型中，时间是按照小时来计算的，运算平台分为Linux/UNIX和Windows两种，并且根据占用资源的数量分为若干等级，各个等级的计价有所不同。通过这种计价方式，用户可以在负载较低的时候选择较小的实例，甚至在空闲的时候停止部分虚拟机的运行。由此可见，这种在类型和时间上更加细粒度的计费模型将有助于用户进一步降低IT成本。

表5.4 Amazon公司提供的配置实例和收费标准

| 类型 | 型号 | 实例配置 | Linux/UNIX系统 | Windows系统 |
|---|---|---|---|---|
| 标准 | 小 | 1.7GB内存，1个EC2计算单元，160GB存储，32位平台 | 0.10美元/小时 | 0.125美元/小时 |
| | 大 | 7.5GB内存，4个EC2计算单元，850GB存储，64位平台 | 0.40美元/小时 | 0.50美元/小时 |
| | 超大 | 15GB内存，8个EC2计算单元，1690GB存储，64位平台 | 0.80美元/小时 | 1.00美元/小时 |
| 高CPU型 | 中 | 1.7GB内存，5个EC2计算单元，350GB存储，32位平台 | 0.20美元/小时 | 0.30美元/小时 |
| | 超大 | 7GB内存，20个EC2计算单元，1690GB存储，64位平台 | 0.80美元/小时 | 1.20美元/小时 |

### 5. 降低管理开销

对于云计算的用户来说，除了降低IT的使用门槛，更重要的是云计算平台还可以帮助他们实现应用的自动化管理。对于应用的运行和管理来讲，云计算的出现能够使用户获得更高的灵活性和自动化。

对应用管理的动态、高效率、自动化是云计算的核心。它要保证用户在创建一个服务的时候，能够用最少的操作和极短的时间就完成资源分配、服务配置、服务上线和服务激活等一系列操作。与此类似，当用户需要停用一个服务的时候，云计算能够自动完成服务停止、服务下线、删除服务配置和资源回收等操作。在虚拟化技术的支持下，Web应用可以被做成虚拟器件，当需要启动服务的时候，被快速部署到云计算环境中；当服务不再需要时，可以取消部署以释放占用的资源。可见，云计算可以在软件和解决方案等不同层次提供极大的灵活性与自动化。

除了应用的部署与删除，在应用的整个生命周期中，时时刻刻需要按照其当前状态进行动态管理，比如根据业务需求增删功能模块、增减资源配置等。在云计算中，这些工作也将在不同程度上由云平台自动完成，为用户提供了灵活的业务管理和便捷的服务。

### 5.2.2 云计算带来的变革

前面我们从IT产业和各个实体的角度分析了云计算的优势，云计算作为一种新兴的IT运用模式，带来了IT产业调整和升级，同时也催生了一条全新的产业链。这条产业链中主要包含硬件供应商、基础软件提供商、云提供商、云服务提供商、应用提供商、企业机构用户和个人用户等不同角色。

如图5.6所示是云计算产业结构中的角色。在云计算的产业结构中，位于中心的是云提供商。云提供商为云服务提供商搭建公有云环境，为企业和机构用户搭建私有云环境。云提供商从硬件提供商和基础软件提供商那里采购硬件和软件，向上提供构建云计算环境所需的解决方案。应用提供商从云服务提供商那里获得所需的资源来开发和运营自己的应用，为个人用户和企业机构用户提供服务。除了从云提供商那里获得私有云，从应用提供商那里获得随时可用的软件外，企业机构用户还可以直接从云服务提供商那里获得计算和存储资源来运行企业机构内部的自有应用。

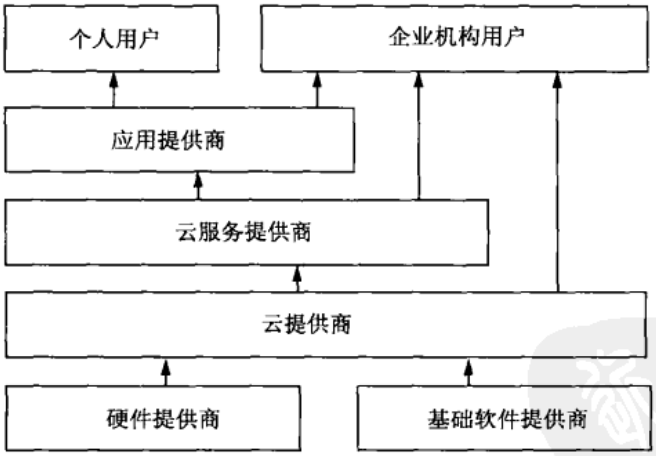



图5.6 云计算产业结构中的角色

云计算将为IT产业带来深刻的变革，也为创业者带来新的机遇。本节将自底向上从这条产业链中的各个角色出发，简要介绍云计算带来的变革。

### 1. 硬件提供商

云计算对当前硬件提供商的业务具有很大的影响。作为硬件的行业客户，一些企业和机构考虑按照云提供商给出的解决方案，增购服务器或者进行技术升级，来构建完全可以由自己控制的私有云环境；也有一些公司将继续以传统的方式使用服务器并且不改变服务器的购买计划。但是，云计算会使得更多的公司，尤其是中小型企业都开始重新考虑甚至放弃原有的服务器购买计划，转而通过使用公有云来提高业务的灵活性，降低运营成本。

然而，这并不意味着云计算会打压硬件提供商的业务。相反，为了满足用户对公有云的需求，云服务提供商将建设更多的公有云环境。这将创造市场对硬件产品的新需求，并促进硬件产品在技术上的创新。那些更加节能、灵活并且能够支持云计算技术要求，尤其是支持虚拟化功能的硬件产品，将在未来的市场中占据更大的份额。

### 2. 基础软件提供商

基础软件包括传统意义上的操作系统和中间件。云计算对于基础软件提供商的影响是巨大的。云计算所带来的变革将影响着从操作系统到上层应用整个软件体系结构的每个角落。在云计算中，互联网就像是一个巨大的操作系统，它运行着云中所有的软件并向用户提供服务。由于越来越多的应用都从桌面操作系统搬到了互联网上，这使得传统操作系统提供商承受着巨大的挑战和压力，一方面必须在新版本的操作系统中引入对云计算核心技术的支持，如虚拟化技术，从而在未来云基础设施领域中占据更多的市场份额；另一方面，如果已有客户要采纳这些新技术，就意味着比较复杂的升级周期，这在从操作系统桌面应用升级到云应用的过程中体现得最为明显。

与操作系统相同，中间件为上层服务提供了通用的功能模块，并且隐蔽了实现细节，使得上层软件的开发可以着重于业务逻辑，而非烦琐的底层细节。在云计算环境中，中间件对上层依然需要提供相同的便捷功能，但是对下层它需要隐藏的细节就更加复杂了。首先，中间件运行在云之上，而不是在传统意义上的单个服务器上，这样它不但需要适应单个云服务提供商的运行环境，而且还要具有跨多个云服务提供商的互操作性。其次，在云上运行的中间件必须支持云计算的核心特征——可扩展性，可以随时随地为任何用户调整资源以满足业务上的需求。可见，作为提供操作

系统和中间件的基础软件提供商，新技术的研发和新产品的推出速度将决定其能否在云计算中占据领先地位。

### 3. 云提供商

云提供商处于云计算产业的核心位置，它向下采购（或者通过咨询服务的方式建议云服务提供商和企业机构用户采购）硬件提供商及基础软件提供商的硬件与软件产品，向上为云服务提供商提供构建公有云的解决方案，为企业机构用户提供构建私有云的解决方案。可见，云提供商在云计算产业中处于“造云者”的角色。可以说，在云计算产业中，其他角色的业务流转都是围绕云提供商展开的。

云提供商需要具有三个显著特点。第一，具有丰富的硬件系统集成经验。云计算无疑将带来现有数据中心的技术升级和扩容，以及新兴大型数据中心的建造。为这些数据中心提供从处理、存储到网络的集成解决方案是一项复杂的系统工程，因此需要云提供商在这方面具有深刻的认识和丰富的经验。第二，具有丰富的软件系统集成经验。硬件是云计算的躯体，软件是云计算的灵魂。从操作系统到中间件，从数据库、Web服务到管理套件，软件的选择、配置与集成方案种类众多、千变万化，如何帮助用户做出最合适的选择，需要云提供商对软件集成具有深刻的理解。第三，具有丰富的行业背景。这一点主要是针对企业机构的私有云建设。由于用户是身处各行各业的不同企业机构，其业务也不尽相同，因此如何为用户设计出最适合自己的私有云解决方案，就需要云提供商对该行业具有深刻的理解和丰富的行业经验。

总之，云提供商需要同时具有丰富的硬件、软件和行业经验才能保证其在云计算产业中的核心位置。云计算产业中的其他角色围绕着云提供商运营流转。云提供商为产业链中的其他角色提供服务，创造价值。

### 4. 云服务提供商

云计算是互联网时代信息技术发展和信息服务需求共同作用下的产物。传统的软件提供商所提供的产品并不能直接适用于云计算环境。规模较小的独立软件提供商一般没有强大的技术实力去实现云计算技术的创新，而规模庞大的专业软件提供商在实现传统软件产品转型时遇到的技术和业务压力也是空前的，这就给那些眼光卓越的精英们带来了创业机会。

这些新兴企业在面对变革时没有沉重的包袱，能够充分而直接地构建

适合互联网时代需求的云计算产品。他们与云提供商紧密合作，提供适合市场需求的云计算环境。无疑，云计算打开了一片宽广的市场空间，无论是基础设施云、平台云还是应用云，都有着巨大的潜在需求。因此，对于每一家云服务提供商，只要能够通过变革和创新来提供便捷的、差异化的云计算服务，他们就能够在云计算产业中获得成功。

### 5. 应用提供商

传统的应用提供商将其应用运行在自己的服务器或者在数据中心中租赁的服务器上，这种传统的方式有着几个弊端。首先，应用提供商要负担更高的成本，因为需要购买或者租赁物理机器，购买相应的各种软件。其次，应用提供商需要对所有的机器和软件进行维护，保证整个系统从硬件到软件都正常地工作。更重要的是，由于成本控制，应用提供商很难用更为低廉的方式获取更多的资源，这会使得服务质量在服务高峰期受到很大影响。

在云计算中，应用提供商所提供的服务运行在云中，并且是以服务的方式通过互联网提供的。云计算能够有效地使应用提供商避免上述弊端，从而为中小企业和刚刚起步的企业降低成本。首先，应用提供商不需要购买专门的服务器硬件及各种软件，只需要将应用部署在云平台中即可，所需的硬件资源和软件服务都由云提供。其次，由于云平台由专人维护，应用提供商也省去了维护费用。另外，云计算中所有的资源都按照具体使用情况付费，从而避免了传统方式中资源空闲所造成的浪费。最后，云平台上的软件都以服务的形式运行，应用提供商在开发新业务的时候能够以较低的成本充分利用云平台所提供的各种服务，从而加速业务上的创新。

### 6. 个人用户

云计算时代将产生越来越多的基于互联网的服务，这些服务丰富全面、功能强大、使用方便、付费灵活、安全可靠，个人用户将从主要使用软件变为主要使用服务。在云计算中，服务运行在云端，用户不再需要购买昂贵的高性能的电脑来运行种类繁多的软件，也不需要对这些软件进行安装、维护和升级，这样可以有效减少用户端系统的成本与安全漏洞。更重要的是，与传统软件的使用方式相比，云计算能够更好地服务于用户。在传统方式中，一个人所能使用的软件仅为其个人电脑上的所有软件。而在云计算中，用户可以通过互联网随时访问不同种类和功能的服务。

云计算将数据放在云端的方式给很多人带来了顾虑，通常人们认为数据只有保存在自己看得见、摸得着的电脑里才最安全，其实不然，因为个人电脑可能会不小心被损坏、遭受病毒攻击，导致硬盘上的数据无法恢复，数据也有可能被木马程序或者有机会接触到电脑的不法之徒窃取或删除，笔记本电脑还存在丢失的风险。而在云环境里，有专业的团队来帮用户管理信息，有先进的数据中心帮助用户备份数据。同时，严格的权限管理策略可以帮助用户放心地与指定的人共享数据。这就如同把钱存到银行里比放在家里更安全一样。

### 7. 企业机构用户

对于一个企业用户来讲，云计算意味着很多。正如上文所述，企业不必再拥有自己的数据中心，大大降低了运营IT部门所需的各种成本。由于云所拥有的众多设备资源往往不是某一个企业所能拥有的，并且这些设备资源由更加专业的团队进行维护，因此企业的各种软件系统可以获得更高的性能和可靠性。另外，企业不需要为每个新业务重新开发新的系统，云中提供了大量的基础服务和丰富的上层应用，企业能够很好地基于这些已有的服务和应用，从而在更短的时间内推出新业务。

当然，也有很多争论说云计算并不适合所有的企业和机构，比如对安全性、可靠性都要求极高的银行、金融企业，还有涉及国家机密的军事单位等，另外如何将现有的系统迁入到云中也是一个难题。尽管如此，很多普通制造业、零售业等类型的企业都是潜在的能够受益于云计算的企业。而且，对于那些对安全性和可靠性要求很高的企业和机构，他们也可以选择在云提供商的帮助下建立自己的私有云。随着云计算的发展，必将有更多的企业用户从不同方面受益于云计算。

## 5.3 云计算产生的原动力

上面我们介绍了云计算的优势，并从云计算产业的角度分析了各个产业参与者将要或者已经面对的变革。实际上，早在1966年D. F. Parkhill在其经典的《计算机效用事业的挑战》（“The Challenge of the Computer Utility”）一书中就大胆预测了一个计算能力如同水和电一样被供给的世界。此后，计算机科学家们向着这个目标不断努力探索，经过多年的挫折与失败，却始终没有一个成功的方案让工业界与市场接受。但是，云计算的出现正在改变这一切。云计算让人们了解到，原来计算、存储和应用也

可以像水和电一样地去获得。

在过去的30年中，我们目睹了发达国家将低端制造业向发展中国家转移，从而完成自身产业升级的全过程。上一节分析了云计算带来的优势，从IT产业的角度出发，云计算顺应了资源合理配置、合理专业化分工的历史潮流。由此，规模效益与全球化分工在IT业界逐渐形成。正如托马斯•弗里德曼在《世界是平的》这本书中所述，分布在世界各地的企业和个人正在由互联网更紧密地联系起来，世界正变得越来越平坦，资源合理配置、专业化分工和规模效益这些原本只在传统制造业中出现的名词已经被应用于IT产业。

可见，云计算带来的是IT产业的转型和升级。不仅各个微观经济实体成为了云计算产业链中的参与者，各国政府也同样重视这一产业的重要变革。毕竟，就如同制造业的变革导致了全球范围内的重新分工，云计算的出现也将引发IT产业在世界范围内的再分工。世界各国，尤其是新兴发展中国家不应错过这个难得的机遇实现自己产业结构的升级。各国政府对于高科技产业的重视程度和投入力度是推动云计算向前发展的重要动力。

在技术层面，云计算之所以在今天产生，是六方面原动力共同作用的结果，如图5.7所示。


图5.7　云计算产生的原动力

第一是芯片和硬件技术的飞速发展，使得硬件能力激增、成本大幅下降，让独立运作的公司集中可观的硬件能力实现规模效益成为可能。

第二是虚拟化技术的成熟，使得这些硬件资源可以被有效地细粒度分

割和管理，以服务的形式提供硬件和软件资源成为可能。

第三是面向服务架构的广泛应用，使得开放式的数据模型和通信标准越来越广泛地为人们使用，为云中资源与服务的组织方式提供了可行的方案。

第四是软件即服务模式的流行，云计算以服务的形式向最终用户交付应用的模式被越来越多的用户所接受。

第五是互联网技术的发展，让网络的带宽和可靠性都有了质的提高，使得云计算通过互联网为用户提供服务成为可能。

第六是Web 2.0技术的流行和广泛接受，改变了人们使用互联网的方式，通过创新的用户体验为云计算培育了使用群。

下面具体介绍这些推动云计算出现和发展的技术原动力。





### 5.3.1 芯片与硬件技术

半导体芯片技术遵循着摩尔定律在不断发展，摩尔定律是指集成电路上可容纳的晶体管数目，约每隔18个月便会增加一倍，性能也将提升一倍。同时计算能力、内存容量、磁盘存储容量也相应地快速提升。多核技术可以在一枚处理器中集成多个完整的计算引擎，它的出现规避了仅仅提高单核芯片的速度而产生过多热量且无法带来相应的性能改善的问题。处理器位数的提高与总线技术的提升，使系统能够支持容量与吞吐量都更大的内存，满足日益增长的应用需求，使更多的任务可以同时运行。随着磁记录技术和机械工艺的不断改进，磁盘的存储容量在增大，数据传输率在提高，寻迹时间在缩短。这些芯片与硬件技术的变革直接作用于计算机系统，使单个系统的能力越来越强，成本越来越低。

除了计算机系统能力的提高，系统间的通信能力也在增强。IEEE 802.3ae定义了带宽为10GB的以太网标准，企业级交换机也支持了10GB全速第二层转发。大量相对廉价的x86系统可以通过高速网络被组织成为大规模的分布式系统，通过协同和冗余来获得以往在大型机上才能达到的处理速度和可靠性。但是，大量地运用廉价系统也带来了这样或那样的问题，如大规模系统难于维护、资源消耗高等。在探索解决这些问题的新技术的过程中，云计算应运而生。

芯片与硬件技术的提升也为数据中心的建造创造了便利条件。伴随着

速度的不断提升，硬件价格也在不断下降。以前，建设大规模数据中心所需的巨大资金投入，只有极少数企业或者政府机构能够负担得起。现在，由于硬件性能的提升和价格的下降，建造大型数据中心已经不再是不可实现的目标。这就为云服务提供商构建公有云，为企业机构用户构建私有云创造了可能。

### 5.3.2 资源虚拟化

在云计算中，数据、应用和服务都存储在云中，云就是用户的超级计算机。因此，云计算要求所有的资源能够被这个超级计算机统一地管理。但是，各种硬件设备间的差异使它们之间的兼容性很差，这为统一的资源管理提出了挑战。

虚拟化技术可以将物理资源等底层架构进行抽象，使得设备的差异和兼容性对上层应用透明，从而允许云对底层千差万别的资源进行统一管理。此外，虚拟化简化了应用编写的工作，使得开发人员可以仅关注于业务逻辑，而不需要考虑底层资源的供给与调度。在虚拟化技术中，这些应用和服务驻留在各自的虚拟机上，有效地形成了隔离，一个应用的崩溃不至于影响到其他应用和服务的正常运行。不仅如此，运用虚拟化技术还可以随时方便地进行资源调度，实现资源的按需分配，应用和服务既不会因为缺乏资源而性能下降，也不会由于长期处于空闲状态而造成资源的浪费。最后，虚拟机的易创建性使应用和服务可以拥有更多的虚拟机来进行容错和灾难恢复，从而提高了自身的可靠性和可用性。

可见，正是由于虚拟化技术的成熟和广泛运用，云计算中计算、存储、应用和服务都变成了资源，这些资源可以被动态扩展和配置，云计算最终在逻辑上以单一整体形式呈现的特性才能实现。虚拟化技术是云计算中最关键、最核心的技术原动力。

### 5.3.3 面向服务架构

面向服务架构（Service Oriented Architecture，SOA）是一种IT架构设计模式，通过这种设计，用户的业务可以被直接转换成为能够通过网络访问的一组相互连接的服务模块。这个网络可以是本地网络或者是互联网。面向服务架构所强调的是将业务直接映射到模块化的信息服务，并且最大

程度地重用IT资产，尤其是软件资产。当使用面向服务架构来实现业务时，用户可以快速创建适合自己的商业应用，并通过流程管理技术来加速业务的处理，促进业务的创新。面向服务架构还可以为用户屏蔽掉运行平台及数据来源上的差异，从而使得IT系统能够以一种一致的方式提供服务。

面向服务架构的设计思想引领了Web服务技术的发展，使得开放式的数据模型和通信标准越来越广泛地为人们使用，更大程度上地促进了已有信息系统的互联。面向服务架构通过基础设施层、业务层、服务层、流程层的层次划分，将模块化的服务和标准化的流程封装成为可以被用户直接应用的组件，允许用户按照自己的实际情况选择、搭建灵活的IT架构，满足业务需求。

资源和功能服务化是云计算的一个核心思想。面向服务架构为云中的资源与服务的组织方式提供了可行的方案。云计算依赖于面向服务架构的思想，通过标准化、流程化和自动化的松耦合组件为用户提供服务。不过，云计算将不仅是一种设计架构的模式或方法，而且是一个完整的应用运行平台，基于面向服务架构思想构建的解决方案将在云中运行，服务于云外的用户。





### 5.3.4 软件即服务

软件即服务（Software as a Service，SaaS）是一种通过互联网提供软件的服务模式，用户不用再一次性购买软件，而改用向服务提供商租用软件，且无需对软件进行维护，服务提供商会全权管理和维护软件。其核心理念是将软件直接提供为服务，从而改变目前常见的软件销售并安装在客户自己的计算机上的这种消费及使用模型。对于中小型企业来说，SaaS消除了购买、安装和维护基础设施、中间件和应用程序的投资环节。从技术方面来看，企业无需再配备专业技术人员进行管理，同时又能得到最新的技术应用。

此外，SaaS也深刻改变了IT业界的商业模式。“长尾理论”被认为是使SaaS在商业上取得成功的理论基础。长尾理论讲求的是充分发掘那80%的零散但充满潜力的市场。从同样的理论出发，利用软件即服务的思想，云计算可以开发那部分曾经无法拥有专业计算中心和Web应用的客户，尤其是中小企业和初创型公司，为他们提供那些曾经只有实力雄厚的大公司才能够负担得起的IT基础设施和应用。

软件即服务技术是云计算的先行者，比如软件的远程使用、按需付费模式。然而软件即服务提供商一般仅仅提供某一种特定的应用软件。云计算就是把这种单一的模式更广泛推广的技术，其采用的虚拟化等技术使得普通软件也可以成为服务，比如Amazon公司的计算和存储服务就可以适应于企业的更多应用类型。

### 5.3.5 互联网技术

近二十年来，世界各国在互联网基础设施建设方面进行了巨额的投资，互联网的带宽和可靠性都得到了大幅提升，网络的触角所涉及的区域也越来越广。目前，信息技术的发展使得世界上大部分的业务都离不开互联网的支持。互联网成为了让世界运转不可缺少的平台。网上纷繁复杂的业务对于互联网上资源的稳定性、可靠性、安全性、可用性、灵活性、可管理性、自动化程度甚至节能环保等特性都提出了苛刻的要求，这一切都在不断推动着互联网技术的发展。正是由于互联网的发展，使得云计算中跨地域的资源共享与服务提供成为可能。

除了骨干网的发展，互联网的接入方式也发生了质的转变。从PSTN拨号上网到ADSL宽带上网，从单一的有线连接到灵活的无线接入，从高速而廉价的WiFi到潜力巨大的3G，从单一的计算机接入到手机、汽车及各种家用电器的接入，可以说，互联网已经是随时随处可用了。不论是在办公室、在家，还是在路途中，稳定的互联网接入是用户获取云计算中丰富多彩资源的基础，不断提高的带宽是用户获得完美体验的前提。正是由于互联网接入的普及和改善，使得用户通过互联网使用远程云端的服务成为可能，在用户和云间搭起了宽阔的桥梁。

### 5.3.6 Web 2.0技术

今天，Web 2.0已经成为了实际意义上的标准互联网运用模式。以博客（Blog）、内容聚合（RSS）、百科全书（Wiki）、社会网络（SNS）和对等网络（P2P）为代表的Web 2.0应用已经被用户广泛地接受和使用。Web 2.0的出现让用户从信息的获得者变成了信息的贡献者，也让富互联网应用（Rich Internet Application，RIA）成为网络应用的发展趋势。例如，Ajax是支持RIA的编程框架，帮助RIA在客户端实现友好

而丰富的用户体验。在该框架中，HTML和CSS为信息提供静态表述，JavaScript负责信息的动态呈现及信息与用户的交互。在Ajax中，浏览器和服务器之间的交互是异步的，这样就避免了页面被重复刷新，从而实现了类似于本地程序的用户体验。

Web 2.0的出现和广泛流行深刻地影响了用户使用互联网的方式。现在，人们越来越习惯从互联网上获得所需的应用与服务，同时将自己的数据在网络上共享与保存。而以往，这些都是用户在个人电脑上完成的工作。个人电脑渐渐不再是为用户提供应用、保存用户数据的中心，它蜕变成为接入互联网的终端设备。Web 2.0提供了云计算的接入模式，也为云计算培养了用户习惯。

Web 2.0为云计算的出现提出了内在需求。随着Web 2.0的产生和流行，互联网用户更加习惯将自己的数据在网络上存储和共享。每天，视频网站和图片共享网站都要接受海量的上载数据。同时，为了给用户提供新颖而有吸引力的服务，Web应用的开发周期越来越短，只有更加快捷的业务响应才能让应用提供商在激烈的竞争中生存。因此，他们需要有这样一个平台，能够提供充足的资源保证其业务增长，能够提供可以复用的功能模块来保证其快速开发。这些，都是云计算产生的内在需求。

## 5.4 小结

本章从4个典型案例出发，介绍了云计算的概念与分类，分析了云计算的特征，并将云计算与其他相关概念进行了辨析。随后，我们从产业到技术，首先在IT产业的层面，从合理配置资源和专业分工的角度分析了云计算的优势，再逐渐深入到云计算的实体层面，从技术革新及提高效益的角度分析了云计算的优势。然后，我们从云计算催生的产业链的角度出发，分析了云计算为这条产业链上每一类参与者带来的深刻变革，以及为创业者带来的新的机遇。最后，我们为读者解析了云计算产生的原动力。在随后的第6章，我们将深入技术层面，分析云计算的架构。在第7章，我们将介绍云环境构建的关键技术。在第8章，我们将介绍云计算的最新的业界动态。

CHAPTER 6

# 第6章 云架构

作为一种新兴的计算模式，云计算能够将各种各样的资源以服务的方式通过网络交付给用户。这些服务包括种类繁多的互联网应用、运行这些应用的平台，以及虚拟化后的计算和存储资源。与此同时，云计算环境还要保证所提供服务的可伸缩性、可用性与安全性。云计算需要清晰的架构来实现不同类型的服务及满足用户对这些服务的各种需求。本章将介绍典型的云架构的基本层次，以及各个层次的功能。另外，我们还通过典型的服务示例来帮助读者更好地理解云架构。



## 6.1 概述

### 6.1.1 云架构的基本层次



通过上一章我们了解到，云计算中的云分为基础设施云、平台云和应用云。这样的分类方式其实已经包含了云架构的基本层次。云架构通过虚拟化、标准化和自动化的方式有机地整合了云中的硬件和软件资源，并通过网络将云中的服务交付给用户。典型的云架构分为三个基本层次：基础设施（Infrastructure）层、平台层（Platform）和应用（Application）层，如图6.1所示。在上一章描述的按云服务类型分类的方式就是按照在云架构不同层次上提供服务的维度来划分的。从图6.1中我们可以发现，这三种层次向上提供服务的方式有公有云、私有云和混合云三种类型，这也正是上一章介绍的云计算按其提供方式所划分的类别。

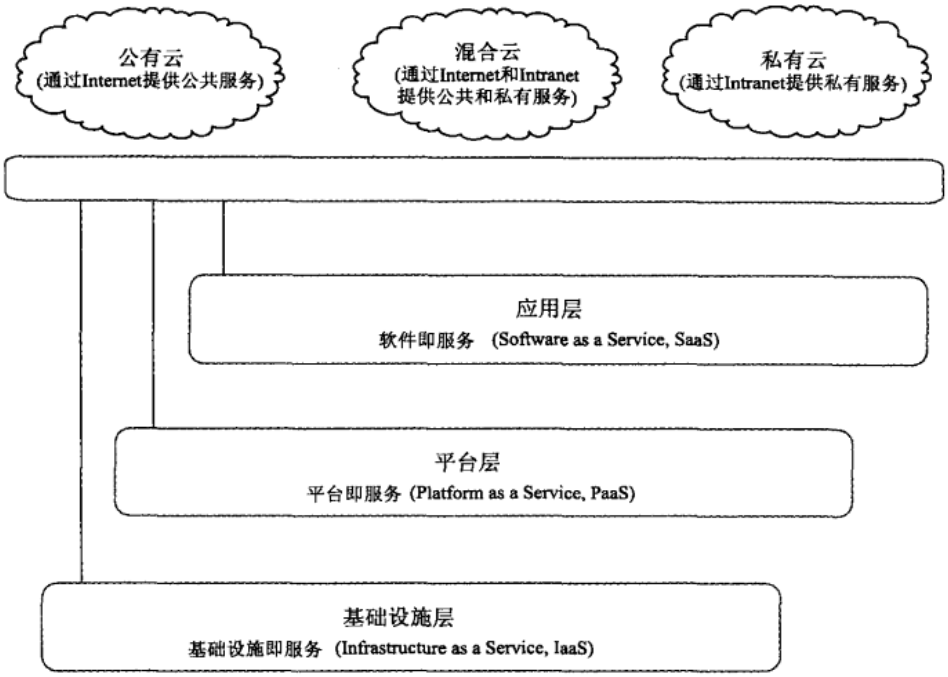



图6.1 云架构层次示意图

基础设施层是经过虚拟化后的硬件资源和相关管理功能的集合。云的硬件资源包括了计算、存储和网络等资源。基础设施层通过虚拟化技术对这些物理资源进行抽象，并且实现了内部流程自动化和资源管理优化，从而向外部提供动态、灵活的基础设施层服务。

平台层介于基础设施层和应用层之间，它是具有通用性和可复用性的软件资源的集合，为云应用提供了开发、运行、管理和监控的环境。平台层是优化的“云中间件”，能够更好地满足云的应用在可伸缩性、可用性和安全性等方面的要求。

应用层是云上应用软件的集合，这些应用构建在基础设施层提供的资源和平台层提供的环境之上，通过网络交付给用户。云应用种类繁多，既可以是受众群体庞大的标准应用，也可以是定制的服务应用，还可以是用户开发的多元应用。第一类主要满足个人用户的日常生活办公需求，比如文档编辑、日历管理、登录认证等；第二类主要面向企业和机构用户的可定制解决方案，比如财务管理、供应链管理和客户关系管理等领域；第三类是由独立软件开发商或开发团队为了满足某一类特定需求而提供的创新型应用，一般在公有云平台上搭建。

需要注意的是，并不是所有的云都必须在这三个层次上分别提供服务。如上一章中介绍的Amazon EC2、Google App Engine和Salesforce

CRM，它们就只分别向用户交付基础设施层、平台层和应用层上的服务。对于云提供商来说，交付的层次越高，其内部需要实现的功能就越多。例如，Amazon EC2为用户提供的是虚拟化的硬件资源，并提供对这些资源的管理；Google App Engine除了需要对硬件资源进行抽象和管理外，还要为用户提供统一的应用开发和运行环境；对于Salesforce CRM，不仅要提供对底层硬件和上层软件平台的支持，还要为用户开发立即可用的软件或软件功能模块。可见，位于云架构上层的云提供商在为用户提供该层的服务时，同时要实现该架构下层所必须具备的功能。虽然实现的方法和细节不尽相同，如Salesforce.com与Amazon可以采用不同的硬件抽象方法，但是这些必备功能是使其服务可以被称之为“云”的必要元素。

## 6.1.2 云架构的服务层次

在前一小节我们提到，云架构中的每一层都可以为用户提供服务，进而出现了基础设施即服务（Infrastructure as a Service，IaaS）、平台即服务（Platform as a Service，PaaS）和软件即服务（Software as a Service，SaaS）的概念。在本小节中，我们将介绍这些服务来使读者进一步了解云架构。

### 1. 基础设施即服务

基础设施即服务交付给用户的是基本的基础设施资源。用户无需购买、维护硬件设备和相关系统软件，就可以直接在基础设施即服务层上构建自己的平台和应用。基础设施向用户提供了虚拟化的计算资源、存储资源和网络资源。这些资源能够根据用户的需求进行动态分配。相对于软件即服务和平台即服务，基础设施即服务所提供的服务都比较偏底层，但使用也更为灵活。

Amazon EC2是基础设施即服务的典型实例。它底层采用Xen虚拟化技术，以Xen虚拟机的形式向用户动态提供计算资源。除Amazon EC2的计算资源外，Amazon公司还提供简单存储服务（Simple Storage Service，S3）等多种IT基础设施服务。虽然Amazon EC2的网络资源拓扑结构是公开的，但是其内部细节对用户是透明的，因此用户可以方便地按需使用虚拟化资源。Amazon EC2向虚拟机提供动态IP地址，并且具有相应的安全机制来监控虚拟机节点间的网络，限制不相关节点间的通信，从而保障了用户通信的私密性。从计费模式来看，EC2按照用户使用资源的数量和时间计费，

具有充分的灵活性。

### 2. 平台即服务

平台即服务交付给用户的是丰富的“云中间件”资源，这些资源包括应用容器、数据库和消息处理等。因此，平台即服务面向的并不是普通的终端用户，而是软件开发人员，他们可以充分利用这些开放的资源来开发定制化的应用。

在平台即服务上开发应用和传统的开发模式相比有着很大的优势。

第一，由于平台即服务提供的高级编程接口简单易用，因此软件开发人员可以在较短时间内完成开发工作，从而缩短应用上线的时间。

第二，由于应用的开发和运行都是基于同样的平台，因此兼容性问题较少。

第三，开发者无需考虑应用的可伸缩性、服务容量等问题，因为平台即服务都已提供。

第四，平台层提供的运营管理功能还能够帮助开发人员对应用进行监控和计费。

Google公司的Google App Engine是典型的平台即服务实例。它向用户提供了Web应用开发平台。由于Google App Engine对Web应用无状态的计算和有状态的存储进行了有效的分离，并对Web应用所使用的资源进行了严格的分配，因此使得该平台上托管的应用具有很好的自动可伸缩性和高可用性。

### 3. 软件即服务

软件即服务交付给用户的是定制化的软件，即软件提供方根据用户的需求，将软件或应用通过租用的形式提供给用户使用。软件即服务主要有以下三个特征。

第一，用户不需要在本地安装该软件的副本，也不需要维护相应的硬件资源，该软件部署并运行在提供方自有的或第三方的环境中。

第二，软件以服务的方式通过网络交付给用户，用户端只需要打开浏览器或者某种客户端工具就可以使用服务。

第三，虽然软件即服务面向多个用户，但是每个用户都感觉是独自占

有该服务。

这种软件交付模式无论是在商业上还是技术上都是一个巨大的变革。对于用户来说，他们不再需要关心软件的安装和升级，也不需要一次性购买软件许可证，而是根据租用服务的实际情况进行付费，也就是“按需付费”。

对于软件开发者而言，由于与软件相关的所有资源都放在云中，开发者可以方便地进行软件的部署和升级，因此软件产品的生命周期不再明显。开发者甚至可以每天对软件进行多次升级，而对于用户来说这些操作都是透明的，他们感觉到的只是质量越来越完善的软件服务。

另外，软件即服务更有利于知识产权的保护，因为软件的副本本身不会提供给客户，从而减少了反编译等恶意行为发生的可能。Salesforce.com公司是软件即服务概念的倡导者，它面向企业用户推出了在线客户关系管理软件Salesforce CRM，已经获得了非常积极的市场反响。Google公司推出的Gmail和Google Docs等，也是软件即服务的典型代表。



## 6.2 基础设施层

在上一节，我们介绍了云架构的基本层次、各层次的资源抽象及所提供的服务。从本节开始，我们将深入各个层次，介绍各层所必备的功能。可以说，这些功能是一项服务可以称得上是“云服务”的基本要求。此外，我们还将为读者描述各个层次上云服务的典型示例，让读者更加深入地理解云架构及各层的基本功能和服务。下面让我们自下而上，从基础设施层开始介绍。



### 6.2.1 基础设施层的基本功能

基础设施层使经过虚拟化后的计算资源、存储资源和网络资源能够以基础设施即服务的方式通过网络被用户使用和管理。虽然不同云提供商的基础设施层在其所提供的服务上有所差异，但是作为提供底层基础资源的服务，该层一般都具有以下基本功能。

#### 1. 资源抽象

当要搭建基础设施层的时候，首先面对的是大规模的硬件资源，比如通过网络相互连接的服务器和存储设备等。为了能够实现高层次的资源管

理逻辑，必须对资源进行抽象，也就是对硬件资源进行虚拟化。

虚拟化的过程一方面需要屏蔽掉硬件产品上的差异，另一方面需要对每一种硬件资源提供统一的管理逻辑和接口。值得注意的是，根据基础设施层实现的逻辑不同，同一类型资源的不同虚拟化方法可能存在着非常大的差异。目前，存储虚拟化方面主流的技术有IBM SAN Volume Controller、IBM Tivoli Storage Manager (TSM)、Google File System、Hadoop Distributed File System和VMware Virtual Machine File System等。

另外，根据业务逻辑和基础设施层服务接口的需要，基础设施层资源的抽象往往是具有多个层次的。例如，目前业界提出的资源模型中就出现了虚拟机（Virtual Machine）、集群（Cluster）、虚拟数据中心（Virtual Data Center）和云（Cloud）等若干层次分明的资源抽象。资源抽象为上层资源管理逻辑定义了操作的对象和粒度，是构建基础设施层的基础。如何对不同品牌和型号的物理资源进行抽象，以一个全局统一的资源池的方式进行管理并呈现给客户，是基础设施层必须解决的一个核心问题。





### 2. 资源监控

资源监控是保证基础设施层高效率工作的一个关键任务。资源监控是负载管理的前提，如果不能有效地对资源进行监控，也就无法进行负载管理。基础设施层对不同类型的资源监控方法是不同的。对于CPU，通常监控的是CPU的使用率。对于内存和存储，除了监控使用率，还会根据需要监控读写操作。对于网络，则需要对网络实时的输入、输出及路由状态进行监控。

基础设施层首先需要根据资源的抽象模型建立一个资源监控模型，用来描述资源监控的内容及其属性。Amazon公司的CloudWatch是一个提供给用户来监控Amazon EC2实例并负责负载均衡的Web服务，该服务定义了一组监控模型，使得用户可以基于模型使用监控工具对EC2实例进行实时监测，并在此基础上进行负载均衡决策。同时，资源监控还具有不同的粒度和抽象层次。一个典型的场景是对某个具体的解决方案整体进行资源监控。一个解决方案往往由多个虚拟资源组成，整体监控结果是对解决方案各个部分监控结果的整合。通过对结果进行分析，用户可以更加直观地监控到资源的使用情况及其对性能的影响，从而采取必要的操作对解决方案进行调整。

### 3. 负载管理

在基础设施层这样大规模的资源集群环境中，任何时刻所有节点的负载都不是均匀的，如图6.2所示负载平衡前。

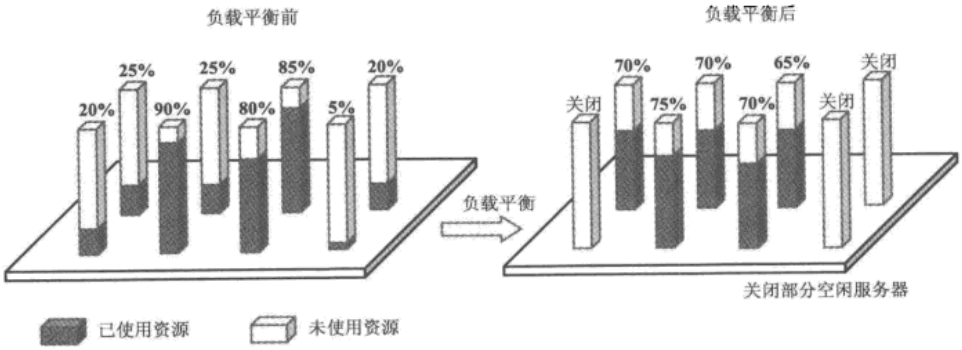



图6.2　负载平衡效果示例

如果节点的资源利用率合理，即使它们的负载在一定程度上不均匀也不会导致严重的问题。可是，当太多节点资源利用率过低或者节点之间负载差异过大时，就会造成一系列突出的问题。一方面，如果太多节点负载较低，会造成资源上的浪费，需要基础设施层提供自动化的负载平衡机制将负载进行合并，提高资源使用率并且关闭负载整合后闲置的资源。另一方面，如果资源利用率差异过大，则会造成有些节点的负载过高，上层服务的性能受到影响，而另外一些节点的负载太低，资源没能充分利用。这时就需要基础设施层的自动化负载平衡机制将负载进行转移，即从负载过高节点转移到负载过低节点，从而使得所有的资源在整体负载和整体利用率上面趋于平衡，如图6.2所示负载平衡后。

### 4. 数据管理

在云计算环境中，数据的完整性、可靠性和可管理性是对基础设施层数据管理的基本要求。现实中软件系统经常处理的数据分为很多不同的种类，如结构化的XML数据、非结构化的二进制数据及关系型的数据库数据等。不同的基础设施层所提供的功能不同，会使得数据管理的实现有着非常大的差异。由于基础设施层由数据中心中大规模的服务器集群所组成，甚至由若干不同数据中心的服务器集群组成，因此数据的完整性、可靠性和可管理性都是极富挑战的。

完整性要求关系型数据的状态在任何时间都是确定的，并且可以通过操作使得数据在正常和异常的情况下都能够恢复到一致的状态，因此完整

性要求在任何时候数据都能够被正确地读取并且在写操作上进行适当的同步。可靠性要求将数据的损坏和丢失的几率降到最低，这通常需要对数据进行冗余备份。可管理性要求数据能够被管理员及上层服务提供者以一种粗粒度和逻辑简单的方式管理，这通常要求基础设施层内部在数据管理上有充分、可靠的自动化管理流程。对于具体云的基础设施层，还有其他一些数据管理方面的要求，比如在数据读取性能上的要求或者数据处理规模的要求，以及如何存储云计算环境中海量的数据等。

### 5. 资源部署

资源部署指的是通过自动化部署流程将资源交付给上层应用的过程，即使基础设施服务变得可用的过程。在应用程序环境构建初期，当所有虚拟化的硬件资源环境都已经准备就绪时，就需要进行初始化过程的资源部署。另外，在应用运行过程中，往往会进行二次甚至多次资源部署，从而满足上层服务对于基础设施层中资源的需求，也就是运行过程中的动态部署。

动态部署有多种应用场景，一个典型的场景就是实现基础设施层的动态可伸缩性，也就是说云的应用可以在极短的时间内根据具体用户需求和服务状况的变化而调整。当用户服务的工作负载过高时，用户可以非常容易地将自己的服务实例从数个扩展到数千个，并自动获得所需要的资源，通常这种伸缩操作不但要在极短的时间内完成，还要保证操作复杂度不会随着规模的增加而增大。另外一个典型场景是故障恢复和硬件维护。在云计算这样由成千上万服务器组成的大规模分布式系统中，硬件出现故障在所难免，在硬件维护时也需要将应用暂时移走，基础设施层需要能够复制该服务器的数据和运行环境并通过动态资源部署在另外一个节点上建立起相同的环境，从而保证服务从故障中快速恢复。

资源部署的方法也会随构建基础设施层所采用技术的不同而有着巨大的差异。使用服务器虚拟化技术构建的基础设施层和未使用这些技术的传统物理环境有很大的差别，前者的资源部署更多是虚拟机的部署和配置过程，而后者的资源部署则涉及了从操作系统到上层应用整个软件堆栈的自动化部署和配置。相比之下，采用虚拟化技术的基础设施层资源部署更容易实现。

### 6. 安全管理

安全管理的目标是保证基础设施资源被合法地访问和使用。在个人电

脑上，为了防止恶意程序通过网络访问计算机中的数据或者破坏计算机，一般都会安装防火墙来阻止潜在的威胁。数据中心也设有专用防火墙，甚至会通过规划出隔离区来防止恶意程序入侵。云计算需要能够提供可靠的安全防护机制来保证云中的数据是安全的，并提供安全审查机制保证对云数据的操作都是经过授权的并且是可被追踪的。

云是一个更加开放的环境，用户的程序可以被更容易地放在云中执行，这就意味着恶意代码甚至病毒程序都可以从云内部破坏其他正常的程序。由于程序在运行和使用资源的方式上都和传统的程序有着较大区别，因此如何在云计算环境里更好地控制代码的行为或者识别恶意代码和病毒代码就成为管理员面临的新挑战。同时，在云计算环境中，数据都存储在云中，如何通过安全策略阻止云的管理人员泄露数据也是一个需要着重考虑的问题。

#### 7. 计费管理

云计算倡导按量计费的计费模式。通过监控上层的使用情况，可以计算出在某个时间段内应用所消耗的存储、网络、内存等资源，并根据这些计算结果向用户收费。对于一个需要传输海量数据的任务，通过网络传输可能还不如将数据存储在移动存储设备中，再由快递公司送到目的地更有效。因为大规模数据传输一方面占用大量时间，另一方面消耗大量网络带宽，数据传输费用相当可观。可见，在具体实施的时候，云计算提供商可以采用一些适当的替代方式来保证用户业务的顺利完成，同时降低用户需要支付的费用。

### 6.2.2 基础设施层服务示例

基于上一小节对基础设施层的介绍，本小节将分析一种基于服务器虚拟化技术的基础设施层简化示例，并在这个示例中简要介绍基础设施层为用户提供的关键服务，以及支持这些服务的基础功能，从而使读者加深对基础设施层的理解。

#### 1. 总体设计

我们在2.2节介绍了服务器虚拟化，服务器虚拟化是一种可以在一台物理服务器上运行多个逻辑服务器的技术，每个逻辑服务器被称为一个虚拟

机。不同的虚拟机之间相互隔离，可以运行不同的操作系统，这使得硬件资源的复用成为可能。服务器虚拟化与其他类型的虚拟化技术，如存储虚拟化、网络虚拟化等，一同奠定了基础设施层进行资源抽象的基础。

下面这个示例所介绍的基础设施层基于虚拟化技术，如图6.3所示。在构建基础设施层前，数据中心的服务器、存储、网络等设备都已准备就绪。完成基础设施层的构建以后，数据中心的硬件设备被整合为虚拟的资源池。管理模块实现了基础设施层的基本功能，这些功能以基础设施即服务的方式提供给用户，用户可以通过这些服务在更高的层次使用基础设施资源。

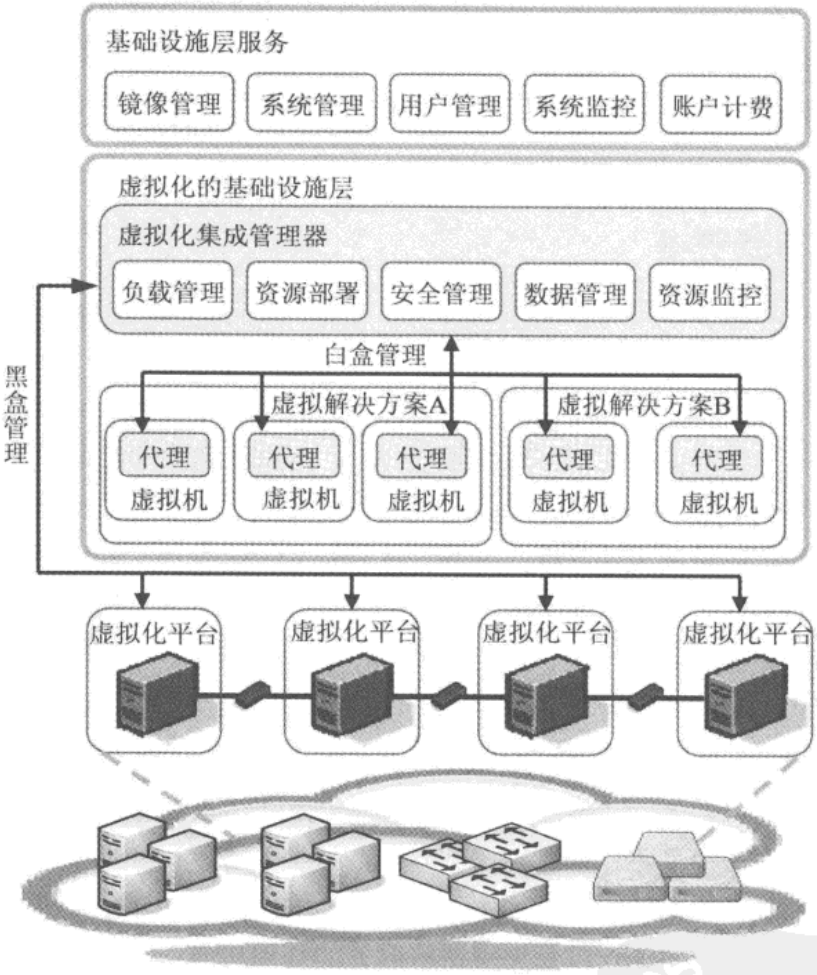



图6.3　基础设施层示例

虚拟化集成管理器是本示例中的基础设施层管理模块，管理的最小单元是虚拟机，它能够完成数据管理、资源监控、负载管理、资源部署、安全管理等功能。另外，该管理器还能够调用虚拟化平台提供的接口，管理

虚拟的硬件资源。

基础设施层服务主要包括镜像管理、系统管理、用户管理、系统监控和账户计费。这些服务与虚拟化集成管理器提供的功能相对应，是用户获得基础设施层资源的接口。下面，我们将通过一个使用基础设施层的典型流程，来向读者介绍该层次提供的各种类型服务，以及支持这些服务的功能。

### 2. 服务流程

典型的基础设施层服务应用流程分为规划、部署和运行三个阶段，如图6.4所示。在规划阶段，基础设施层对硬件资源进行虚拟化，使其成为一个逻辑的资源池，并且配置安全管理模块，控制用户对资源池的访问。基础设施层还要具备对数据尤其是对虚拟镜像文件的管理功能，同时提供给用户访问这些镜像或者上传自定义镜像的服务。在部署阶段，基础设施层实现自动部署资源的功能，从而支持用户通过系统管理服务进行系统部署和卸载。部署阶段过后就进入了运行阶段。在这个阶段，用户的系统已经运行在基础设施层提供的虚拟资源上了。此时，基础设施层需要持续地对该系统进行资源监控、负载管理和安全管理，同时为用户提供系统运行状态监控和账户计费服务。下面我们以这三个阶段为线索，为读者介绍基础设施层的基本功能和其为用户提供的服务。

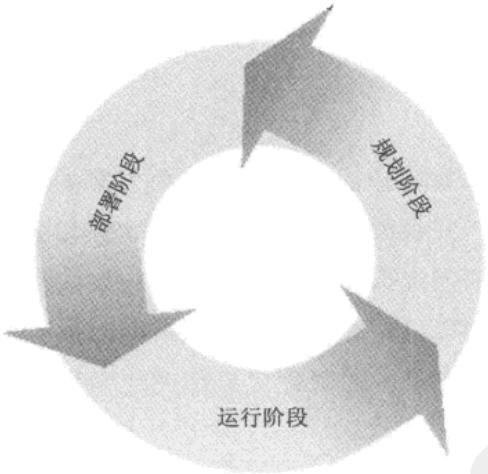



图6.4　基础设施层服务应用流程

### 3. 规划阶段

在基础设施层的物理环境已经准备就绪的状态下，第一个要实现的基本功能就是对资源进行虚拟化的抽象表示。在本示例中，硬件资源的虚拟

化采用的是虚拟化软件，将物理服务器改造成为虚拟化平台，从而整合了计算资源。在此基础上，虚拟化集成管理器通过虚拟化平台提供的接口，获得各种资源的信息，对该平台上的虚拟机进行操作。

该示例中为了使用户能够访问这些虚拟资源，基础设施层允许用户从远程获取资源。用户需要下载一个用户端程序，该程序包含了对基础设施层的访问逻辑，以及保证通信安全的证书和密钥。用户通过这个程序获取现有资源列表，选择其所需要的虚拟机类型，以及进行部署和运行等操作。

该示例中虚拟化集成管理器的数据管理包括两个方面：第一，对业务数据的管理；第二，对虚拟镜像文件的管理。对于业务数据，可以采用传统的数据管理方法。但是，由于镜像文件是二进制数据，虽然大小在10GB～20GB，但是一般镜像文件中包含了虚拟机数据的空间并不多，大部分都是空白。如果没有很好的镜像管理功能，会造成物理存储空间的极大浪费。虚拟化集成管理器的镜像管理一方面通过压缩的方式存储镜像文件，另一方面通过增量备份的方法减少镜像文件的冗余度。

该示例中的虚拟镜像文件包含虚拟机配置、操作系统类型及其上软件堆栈等信息。一个可配置镜像文件模板可以被不同的用户重复使用，基础设施层提供给用户获取已有镜像的服务。如果用户有特殊的需求，现有的镜像文件无法满足其功能需要，基础设施层提供镜像上传服务，允许用户将兼容的镜像进行上载部署。

### 4. 部署阶段

资源部署主要是指虚拟机或者虚拟解决方案的部署，在部署的过程中为虚拟机分配资源，并且激活虚拟机内部的软件和服务。每个虚拟机都有一个配置文件用来描述虚拟机的资源配置，比如内存大小和网络地址。通过虚拟化平台的管理接口，虚拟机及其网络可以被有效地部署，并处于运行状态。然而，虚拟化平台的管理接口却无法为我们激活虚拟机内部的软件，比如中间件产品。

在本示例中，虚拟机内部的代理（Agent）根据OVF文件对虚拟机内部软件的配置描述，激活这些软件。比如虚拟机内部安装了一个应用服务器，同时使用OVF文件描述这个应用服务器实例的配置。当这个虚拟机被部署了以后，虚拟机内部的Agent接收到虚拟化集成管理器的激活指令，根据OVF描述，启动和配置这个实例，使它进入运行状态。

基础设施层在虚拟化集成管理器与OVF描述文件的帮助下实现了解决方案部署的高度自动化，用户端的系统激活逻辑被大大简化。基础设施层提供给用户可视化的OVF文件编辑界面，允许用户根据自己的需求对解决方案进行配置。此后的部署激活工作就如同点击“开始”按钮一样简单。

### 5. 运行阶段

为了能够对虚拟机进行细粒度的运行时管理，在本实例中需要在每个虚拟机内部安装一个代理，如图6.3所示。这个代理负责与虚拟化集成管理器通信，从而实现对虚拟机内部软件的管理。虚拟化集成管理器以两种方式对每个虚拟机进行管理。（1）黑盒式管理：这种管理主要是针对虚拟机整体进行的管理，与虚拟机内部运行什么软件无关，比如虚拟机的内存调整等，这种方式是通过虚拟化集成管理器与虚拟服务器的直接交互完成的。（2）白盒式管理：这种管理主要是对虚拟机内部软件栈进行的管理，比如中间件的监控和配置等，这种方式的管理是通过虚拟化集成管理器与虚拟机内部的代理之间的通信来完成的。

资源监控是通过虚拟化集成管理器的黑盒管理和白盒管理共同完成的。在黑盒管理中，虚拟化集成管理器通过与虚拟服务器的通信，获得每个虚拟机运行时间的资源监控信息。通过对单个虚拟机资源监控信息的进一步分析整合，虚拟化集成管理器还可以计算出整个虚拟解决方案的资源监控信息。在白盒管理中，虚拟化集成管理器需要管理的是虚拟机内部的软件栈。代理负责接收虚拟化集成管理器的状态监控指令，根据该指令监控信息并获取虚拟机内部软件的运行状况监控信息，然后将这些监控信息发给虚拟化集成管理器。值得注意的是，这种白盒管理方式的监控需要被监控的产品支持代理的监控接口标准，从而使得代理能够独立于任何产品。对于一个具体的产品而言，对代理监控接口标准的支持可以是产品自身提供的，也可以由第三方软件提供商支持。这种接口标准具有透明性，而这种透明性正是代理需要为虚拟化集成管理器提供的特性。

负载管理是基于资源监控功能来实现的，并且同样依赖于虚拟化集成管理器的黑盒管理机制和白盒管理机制。在黑盒管理方式下，虚拟化集成管理器根据收集到的监控信息，通过资源调整和资源整合的方式进行负载管理。当虚拟机所在的物理服务器上还有可用资源的时候，可以通过调用虚拟服务器的接口为虚拟机调整存储、内存等各种资源；当虚拟机所在的物理服务器上的可用资源不足时，可以通过虚拟机的实时迁移来进行资源

整合，从而平衡不同服务器之间的负载。在白盒管理方式下，虚拟化集成管理器分析代理发出的监控信息，并将最后的动作指令发给代理，代理执行这些指令并将结果返回给虚拟化集成管理器。由此可见，代理在白盒管理中承担了虚拟化集成管理器与虚拟机内部软件监控管理的桥梁，是白盒管理中的核心模块。

安全管理贯穿于整个运行阶段，不同层次的安全管理对于整个基础设施层的安全都非常重要。首先需要保护的就是虚拟化平台的管理域（如Xen的Domain 0）。一般保护管理域的措施包括在管理域中只运行必要的服务、用防火墙控制对管理域的访问和禁止用户访问管理域等。对于本小节介绍的简化的基础设施层示例来说，虚拟化集成管理器和代理的安全管理至关重要。对它们的访问需要通过安全认证，并且服务的消息中需要包含安全认证信息，从而对所有的访问进行有效的跟踪和记录。在虚拟机内部，不同软件的安全管理对于解决方案的安全同样重要，比如数据库的安全配置会影响到业务数据的安全性。虚拟化集成管理器和代理的安全管理可以与虚拟机内部软件的安全管理相结合，从不同层次对服务和数据的访问进行控制，从而保证云基础设施层的安全。

资源监控和负载管理是为用户提供账户计费、运行状态监控服务的基础。通过对虚拟机的配置、使用时间、负载管理复杂程度及服务质量的综合考虑，基础设施层为用户提供精确的账户计费服务。此外，虽然基础设施层实现了负载管理的自动化，但是用户仍希望获知自己系统的实时状态与历史信息，而运行状态监控服务就满足了用户的这个需求。通过日志信息和统计图表，用户可以了解系统详情，并根据这些信息做出决策，对系统的运行进行必要的手动优化。

## 6.3 平台层

开发、运行和维护是软件生命周期的几个关键环节。虽然目前已经有相应的辅助工具来提高软件开发速度、自动化测试流程、加速版本迭代，但是整个软件周期相对于动态变化的业务需求而言仍显得格外漫长。云计算的出现有望加速产品、服务和解决方案的交付速度。云架构中的平台层负责为用户的应用提供开发、运行和运营环境，同时满足该应用的业务动态需求，为其按需地提供底层资源的伸缩。使用云架构平台层的用户通常是独立软件提供商（Independent Software Vendor，ISV），ISV拥有专业的

开发和运营团队，借助平台层提供的资源，为最终用户提供服务。本节将介绍平台层的基本功能和示例，帮助读者更好地了解云平台如何加速云应用的开发与交付。

### 6.3.1 平台层的基本功能

云计算平台层与传统的应用平台在所提供的服务方面有很多相似之处。传统的应用平台，如本地Java环境或.Net环境都定义了平台的各项服务标准、元数据标准、应用模型标准等规范，并为遵循这些规范的应用提供了部署、运行和卸载等一系列流程的生命周期管理。云计算平台层是对传统应用平台在理论与实践上的一次升级。这种升级给应用的开发、运行和运营各个方面都带来了变革。平台层需要具备一系列特定的基本功能，才能满足这些变革的需求。

#### 1. 开发测试环境

平台层对于在其上运行的应用来说，首先扮演的是一个开发平台的角色。一个开发平台需要清晰地定义应用模型，具备一套API代码库，提供必要的开发测试环境。

一个完备的应用模型包括开发应用的编程语言、应用的元数据模型，以及应用的打包发布格式。一般情况下，平台层基于对传统应用平台的扩展而构建，因此应用可以使用流行的编程语言进行开发，如Google App Engine目前支持Python和Java这两种编程语言。即使平台层具有特殊的实现架构，开发语言也应该在语法上与现有编程语言尽量相似，从而缩短开发人员的学习时间，如Salesforce.com使用的是自有编程语言Apex，该语言在语法和符号表示上与Java类似。元数据在应用与平台层之间起着重要的接口作用，比如平台层在部署应用的时候需要根据应用的元数据对其进行配置，在应用运行时也会根据元数据中的记录为应用绑定平台层服务。应用的打包格式需要指定应用的源代码、可执行文件和其他不同格式的资源文件应该以何种方式进行组织，以及这些组织好的文件如何整合成一个文件包，从而以统一的方式发布到平台层。

平台层所提供的代码库（SDK）和其API对于应用的开发至关重要。代码库是平台层为在其上开发应用而提供的统一服务，如界面绘制、消息机制等。定义清晰、功能丰富的代码库能够有效地减少重复工作，缩短开

发周期。传统的应用平台通常提供自有的代码库，使用了这些代码库的应用只能在此唯一的平台上运行。在云计算中，某一个云提供商的平台层代码库可以包含由其他云提供商开发的第三方服务，这样的组合模式对用户的应用开发过程是透明的。如图6.5所示，假设某云平台提供了自有服务A与B，同时该平台也整合了来自第三方的服务D。那么，对于用户来说，看到的是该云平台提供的A、B和D三种服务程序接口，可以无差异地使用它们。可见，平台层作为一个开发平台应具有更好的开放性，为开发者提供更丰富的代码库和API。

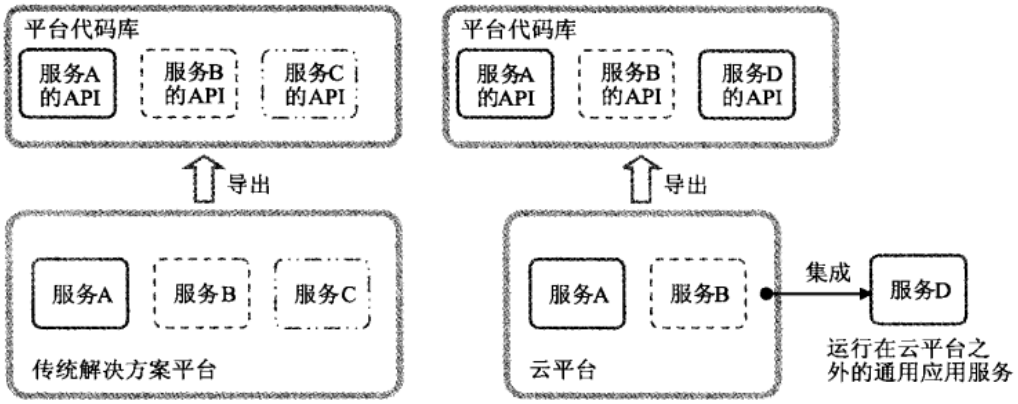



图6.5　传统解决方案平台与云平台的代码库提供方式

平台层需要为用户提供应用的开发和测试环境。通常，这样的环境有两种实现方式。一种方式是通过网络向软件开发者提供一个在线的应用开发和测试环境，也就是说一切的开发测试任务都在服务器端完成。这样做的一个好处是开发人员不需要安装和配置开发软件，但需要平台层提供良好的开发体验，而且要求开发人员所在的网络稳定且有足够的带宽。另外一种方式是提供离线的集成开发环境，为开发人员提供与真实运行环境非常类似的本地测试环境，支持开发人员在本地进行开发与调试。这种离线开发的模式更符合当前大多数开发人员的经验，也更容易获得良好的开发体验。在开发测试结束以后，开发人员需要将应用上传到云中，让它运行在平台层上。

### 2. 运行时环境

完成开发测试工作以后，开发人员需要做的就是对应用进行部署上线。应用上线首先要将打包好的应用上传到远程的云平台上。之后，云平台通过解析元数据信息对应用进行配置，使应用能够正常访问其所依赖的平台服务。平台层的不同用户之间是完全独立的，不同的开发人员在创建

应用的时候不可能对彼此应用的配置和他们将如何使用平台层进行提前约定，配置冲突可能导致应用不能正确运行。因此，在配置过程中需要加入必要的验证步骤，以避免冲突的发生。配置完成之后，将应用激活即可使应用进入运行状态。

以上云应用的部署激活是平台层的基本功能。此外，该层还需要具备更多的高级功能来充分利用基础设施层提供的资源，通过网络交付给客户高性能、安全可靠的应用。为此，平台层与传统的应用运行环境相比，必须具备三个重要的特性：隔离性、可伸缩性和资源的可复用性。

隔离性具有两个方面的含义，即应用间隔离和用户间隔离。应用间隔离指的是不同应用之间在运行时不会相互干扰，包括对业务和数据的处理等各个方面。应用间隔离保证应用都运行在一个隔离的工作区内，平台层需要提供安全的管理机制对隔离的工作区进行访问控制。用户间隔离是指同一解决方案不同用户之间的相互隔离，比如对不同用户的业务数据相互隔离，或者每个用户都可以对解决方案进行自定义配置而不影响其他用户的配置。多租户技术是云计算环境中实现用户间隔离的重要技术，本书将在第7章介绍多租户技术。

可伸缩性是指平台层分配给应用的处理、存储和带宽能够根据工作负载或业务规模的变化而变化，即工作负载或业务规模增大时，平台层分配给应用的处理能力能够增强；当工作负载或者业务规模下降时，平台层分配给应用的处理能力可以相应减弱。比如，当应用需要处理和保存的数据量不断增大时，平台层能够按需增强数据库的存储能力，从而满足应用对数据存储的需求。可伸缩性对于保障应用性能、避免资源浪费都是十分重要的。

云计算平台层是能够容纳数量众多的不同应用的通用平台。该平台的一个重要特性是要满足应用的扩展性。当应用业务量提高，需要更多的资源时，它可以向平台层提出请求，让平台层为它分配更多的资源。当然，这并不是说平台层所拥有的资源是无限的，而是通过统计复用的办法使得资源足够充裕，能够保证应用在不同负载下可靠运行，使其感觉平台层仿佛拥有的资源是无限的，它可以随时按需索取。这一方面需要平台层所能使用的资源数量本身是充足的，另一方面需要平台层能够高效利用各种资源，对不同应用所占有的资源根据其工作负载的变化来进行实时动态的调整和整合。

### 3. 运营环境

随着业务和客户需求的变化，开发人员往往需要改变现有系统从而产

生新的应用版本。云计算环境简化了开发人员对应用的升级任务，因为平台层提供了升级流程自动化向导。为了提供这一功能，云平台要定义出应用的升级补丁模型及一套内部的应用自动化升级流程。当应用需要更新时，开发人员需要按照平台层定义的升级补丁模型制作应用升级补丁，使用平台层提供的应用升级脚本上传升级补丁、提交升级请求。平台层在接收到升级请求后，解析升级补丁并执行自动化的升级过程。应用的升级过程需要考虑两个重要问题：一个问题是升级操作的类型对应用可用性的影响，即在升级过程中客户是否还可以使用老版本的应用处理业务；另一个问题是升级失败时如何恢复，即如何回滚升级操作对现有版本应用的影响。

在应用运行过程中，平台层需要对应用进行监控。一方面，用户通常需要实时了解应用的运行状态，比如应用当前的工作负载及是否发生了错误或出现异常状态等。另一方面，平台层需要监控解决方案在某段时间内所消耗的系统资源。不同目的的监控所依赖的技术是不同的。对于应用运行状态的监控，平台层可以直接检测到诸如响应时间、吞吐量和工作负载等实时信息，从而判断应用的运行状态。比如，可以通过网络监控来跟踪不同时间段内应用所处理的请求量，并由此来绘制工作负载变化曲线，并根据相应的请求响应时间来评估应用的性能。

对于资源消耗的监控，可以通过调用基础设施层服务来查询应用的资源消耗状态，这是因为平台层为应用分配的资源都是通过基础设施层获得的。比如通过使用基础设施层服务为某应用进行初次存储分配。在运行时，该应用同样通过调用基础设施层服务来存储数据。这样，基础设施层记录了所有与该应用存储相关的细节，供平台层查询。

用户所需的应用不可能是一成不变的，市场会随着时间推移不断改变，总会有一些新的应用出现，也会有老的应用被淘汰。平台层需要提供卸载功能帮助用户淘汰过时的应用。平台层除了需要在卸载过程中删除应用程序，还需要合理地处理该应用所产生的业务数据。通常，平台层可以按照用户的需求选择不同的处理策略，如直接删除或备份后删除等。平台层需要明确应用卸载操作对用户业务和数据的影响，在必要的情况下与客户签署书面协议，对卸载操作的功能范围和工作方式做出清楚说明，避免造成业务上的损失和不必要的纠纷。

平台层运营环境还应该具备统计计费功能。这个计费功能包括了两个方面。一方面是根据应用的资源使用情况，对使用了云平台资源的ISV计

费，这一点我们在基础设施层的资源监控功能中有所提及。另一个方面是根据应用的访问情况，帮助ISV对最终用户进行计费。通常，平台层会提供诸如用户注册登录、ID管理等平台层服务，通过整合这些服务，ISV可以便捷地获取最终用户对应用的使用情况，并在这些信息的基础上，加入自己的业务逻辑，对最终用户进行细粒度的计费管理。

## 6.3.2 平台层服务示例

在介绍了平台层的基本功能之后，本小节将结合具体示例进一步介绍平台层。该示例中的平台层运行于上一节介绍的基础设施层之上，为开发人员提供了支持离线开发的SDK和集成开发测试环境，实现了客户应用的自动部署和扩展，并为用户提供了所需的运营环境。通过这个示例，我们将了解平台层如何实现基本功能，以及这些基本功能是如何在一起相互协作的。

### 1. 总体设计

如图6.6所示的平台层示例构建在6.2.2小节所介绍的基础设施层示例的基础之上。平台层采用了多租户的系统架构，包括了运行、运营和开发这三个环境及这些环境所提供的一系列平台层服务。


图6.6 平台层服务示例

本示例中采用扩展的J2EE的企业解决方案模型。扩展的部分一方面是为了实现产品的高级功能，另一方面是为了满足平台层系统本身的需求，比如多租户系统架构要求额外的元数据描述等。数据库服务器集群运行在基础设施层的虚拟机里面，每个虚拟机包含一个数据库服务器节点。应用服务器集群也运行在不同的虚拟机里面，每个虚拟机包含一个应用服务器。

同时，应用服务器和数据库服务器集群都采用了多租户技术来进行租户隔离。不同的应用运行在平台层之上，使用每个应用的不同租户的数据都被彼此隔离起来。每个租户可以根据自己的需要对应用程序进行定制化配置，从而满足具体的业务需求。在基础设施层构建好以后，平台层被打包成为虚拟解决方案，调用基础设施层的资源部署功能对平台层进行初始化部署和激活，从而使平台层进入运行状态。

平台层服务是该层为用户提供的服务的集合，主要包括应用上线、应用升级、应用监控、计费管理、应用开发和应用测试等。这些服务与平台层的基本功能相对应，是用户获得平台层服务的接口。下面我们将通过一个使用平台层服务的典型流程，来向读者介绍该层次提供的各种类型服务，以及支持这些服务需要的功能。





### 2. 平台层提供的环境

平台层提供了开发、运营与运行三个环境，如图6.7所示。在开发环境中，开发人员使用平台层提供的SDK和集成开发环境开发应用。同时，平台层为用户提供了一个模拟运行环境，使开发者可以模拟应用在云生产环境上的运行情况，并进行调试。当开发工作完成后，开发人员将应用按照平台层的规范打包，使用SDK中提供的应用上载服务将应用在平台层上部署和激活。运营环境为用户提供应用的上线、升级、维护和下线管理，以及应用运行状态监控和账户计费等服务。运行环境为应用提供运行时环境，保证其可以自动、高效、高性能地运行。运行环境需要持续地监控应用的行为，保证其隔离性与可伸缩性，执行资源监控与分配。下面我们以这三个环境为线索来介绍平台层的基本功能及其为用户提供的服务。

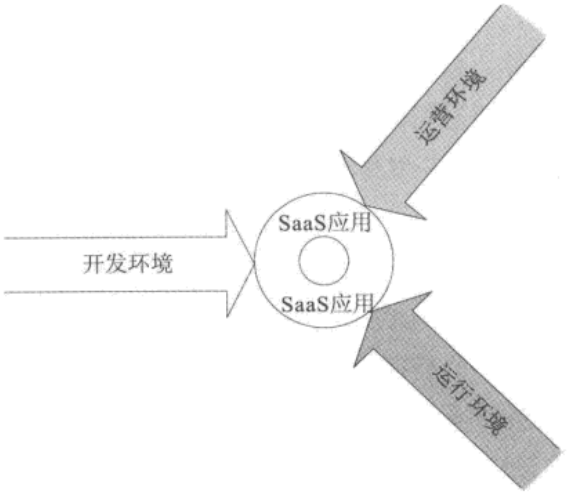



图6.7　平台层提供的环境

### 3. 开发环境

作为一个开发平台，开发环境首先需要定义自己支持的应用模型，本示例采用J2EE规范作为应用模型。开发人员可以采用他们熟悉的语言与开发范式来开发应用。

对于任何一个支持上层应用运行的平台系统来说，它都必须提供清晰的上层应用的模型。模型定义中一个非常重要的问题就是如何描述其将要提供给外界的服务，以及该服务将要以何种方式来提供。无论采用何种方式，云应用中都必须包含对该服务接口的定义，以及描述该服务运行时配置信息的元数据，从而使得平台层能够在云应用部署的时候将该服务变成可用状态。

云应用比较常见的服务提供方式有REST和SOAP方式。REST是Representational State Transfer的简称，它是面向资源的一种软件架构风格，通常对资源的操作有获取、创建、修改和删除。SOAP是Simple Object Access Protocol的简称，它是通过HTTP方式以XML格式交换信息的一种协议，有着完备而复杂的封装机制和编码规则。

为支持应用与平台层的无缝整合，开发环境提供了自己的平台SDK。SDK中包括平台API和通用服务API，以JAR包的形式发布。该SDK能够模拟出与真实运行环境非常相似的测试环境。测试环境应该尽量简单，不需要在架构上完全对生产环境进行复制，只需要保证运行时生产环境中可用的各种服务同样在测试环境中适用，并且所有的使用条件和限制也在测试

环境中被反映出来，比如对磁盘写操作的限制和对网络I/O的限制等。平台层同时开发了插件，使SDK有选择地与流行的集成开发环境（IDE）整合，比如Eclipse等。通过可视化的方式简化整个开发过程中的配置和操作过程，为用户提供良好的开发体验。

### 4. 运营环境

运营环境需要能够有效地处理应用的上线、升级和卸载。这些任务总体可以分为两部分，一个是在J2EE平台上进行升级与卸载的标准流程，另一个是平台层的定制化操作，比如上线需要在运营环境中注册、升级后需要更新运营环境中该应用的版本信息。

在本示例中，为了完成应用上线，开发人员可以使用SDK中的工具对应用进行部署，SDK负责与运营环境进行通信，提交部署请求，进行应用上载操作。然后，运营环境调用其内部的应用部署流程为应用进行数据存储的配置，将其部署在应用服务器上，并在运营环境中注册。最后，应用被激活，进入可用状态。

运营环境为用户提供了管理功能，应用管理员可以通过浏览器访问管理控制台来管理应用的域名及子域名，查询访问记录和错误记录，进行流量分析，设置计划任务，查询账单等。此外，为了实现应用的安全升级，运营环境还为用户提供了一个测试用的沙箱环境。用户将新应用上传至平台层后，可以首先在该沙箱环境中测试应用与平台层的兼容性、业务的正确性与应用的性能等。当一切稳妥后再将其迁出沙箱环境，完成应用的升级。

在监控计费方面，运营环境可以从两个层面进行：一方面是虚拟机层面的，平台层管理器可以利用基础设施层的资源监控能力获得应用运行所占用的存储、内存和CPU等信息；另一方面，云平台层可以利用应用网络传输中的应用标识信息对网络资源的使用进行监控。根据各方面监控的结果有效地计算出应用所使用的资源，并根据计费标准对应用计费。这些信息将统一由平台层管理器获取与管理，通过应用管理控制台呈现给用户。

### 5. 运行环境

本示例中的运行环境需要支持平台层的三个基本特性：隔离性、可伸缩性和拥有无限资源。隔离性依赖于两个方面，一方面是虚拟机之间的系统级别的隔离，另一方面是多租户系统架构所支持的租户级别的隔离。关

于租户级别的隔离，请参照第7章中对多租户技术的介绍。

示例中的平台层通过虚拟化技术可以很好地实现可伸缩性。根据虚拟解决方案中的虚拟机所包含软件的不同，可伸缩性的实现可以采用以下两种基本方案，如图6.8所示。如果应用服务器和数据库服务器是一起打包在一个虚拟机里面的，那么应用服务器和数据库服务器之间已经配置好了，也就是说应用服务器可以直接使用数据库服务器；如果应用服务器和数据库服务器是分别打包的，那么则需要在部署完动态扩展节点之后，由平台层管理器对那些需要更高数据存储能力的应用服务器进行配置，使得应用服务器可以使用这些新扩展的数据库服务器。该平台层基于上一节所述的基础设施层进行构建，可以充分享受虚拟化的基础设施层所带来的资源容量。然而，运营环境仍然需要协调每个虚拟机的资源，使得资源得到高效的使用。

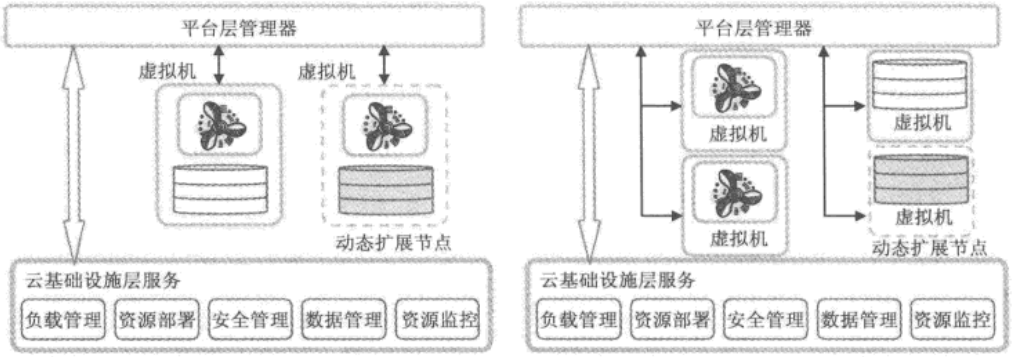



图6.8　平台层可伸缩性示意图

## 6.4 应用层

我们知道，应用层是运行在云平台层上的应用的集合。每一个应用都对应一个业务需求，实现一组特定的业务逻辑，并且通过与用户的交互提供服务。总的来说，应用层的应用可以分为三大类：第一类是面向大众的标准应用，比如Google的文档服务Google Docs、IBM的协作服务LotusLive等；第二类是为了某个领域的客户而专门开发的客户应用，比如Salesforce CRM；第三类是由第三方的独立软件开发商在云计算平台层上开发的满足用户多元化需求的应用，比如礼品清单应用Giftag等。

值得注意的是，不同于基础设施层和平台层，应用层上运行的软件千变万化，新应用层出不穷，想要定义应用层的基本功能十分困难。或者说，应用层的基本功能就是要为用户提供尽可能丰富的创新功能，为企业

和机构用户简化IT流程，为个人用户简化日常生活的方方面面。因此，在本节我们首先为读者总结应用层的特征，再按类别详细介绍每一类应用的典型示例。

### 6.4.1 应用层的特征

应用层是云中应用的集合，回顾本章开始介绍的软件即服务（SaaS）的概念，最终用户就是通过SaaS的方式获得应用层中各种应用服务的。结合SaaS的定义，云计算应用层上的应用需要具有以下三个基本特征。

第一，这些应用能够通过浏览器访问，或者具有开放的API，允许用户或者瘦客户端的调用。云应用的理想模式是不论用户身处何处，不论使用何种终端，只要有互联网连接和标准的浏览器，便可以不经任何配置地访问属于自己的应用。目前，虽然互联网连接速度和Web开发技术已经使基于浏览器的应用具有了非常好的用户体验，但是距离一些在本地安装与运行的软件仍有差距，比如在图形处理方面。因此，在云计算的初期，应用层某些应用也可以通过瘦客户端来访问。这虽然影响了云应用的灵活性，但仍是一种有效的折中方案。

第二，用户在使用云服务时，不需要进行一次性投入，只需要在使用的过程中按照其实际的使用情况付费。首先，用户在使用云服务时不需要购买额外的硬件，因为从处理到数据存储都在云上执行，用户端的处理能力不高也可以访问云上应用。其次，虽然从本质上讲云应用也是供用户使用的软件，但用户不需支付软件副本的费用，只需要注册一个账号，即可开始使用该应用。最后，用户开始使用云应用后，只需按照其实际使用量付费。

第三，云应用要求高度的整合，而且云应用之间的整合能力对于云应用的成功至关重要。云应用之间的整合能力对于完美的用户体验来说是不可或缺的，因为用户的需求往往是综合性的。如果用户所需要的多个功能是由若干个彼此之间无法整合的应用程序来实现的，那么用户体验和操作效率都会不甚理想。由于应用都是运行在云中而且彼此相对独立，因此云应用整合较传统应用会相对容易实现。

### 6.4.2 应用层的分类

上面我们总结了云计算应用层需要具备的三个基本特征。用户不需要

关心应用是在哪里被托管的、是采用何种技术开发的，也不需要在本地安装这些软件，只需要关心如何去访问这些应用。下面我们对常见的云应用进行分类讨论。

第一种类型是标准应用，采用多租户技术为数量众多的用户提供相互隔离的操作空间，提供的服务是标准的、一致的。用户除了界面上的个性化设定外，不具有更深入的自定义功能。可以说，标准应用就是我们常用应用软件的云上版本。可以预见，常用的桌面应用都会陆续出现其云上版本，并最终向云上迁移。

第二种类型是客户应用，该类应用开发好标准的功能模块，允许用户进行不限于界面的深度定制。与标准应用是面向最终用户的立即可用的软件不同，客户应用一般针对的是企业级用户，需要用户进行相对更加复杂的自定义和二次开发。客户应用提供商是传统的企业IT解决方案提供商的云上版本。

第三种类型是多元应用，这类云应用一般由独立软件开发商或者是开发团队在公有云平台上搭建，是满足用户某一类特定需求的创新型应用。不同于标准应用所提供的能够满足大多数用户日常普遍需求的服务，多元应用满足了特定用户的多元化需求。现在，在Google App Engine平台上已经出现了数量众多的多元应用。比如，Mutiny为身处旧金山地区的用户提供了地铁和公交的时刻表服务；The Option Lab为投资者提供了期权交易策略制定、风险分析、收益预期等一揽子方案；FitnessChart帮助正在进行健身练习的用户记录体重、脂肪率等数据，使用户可以跟踪自己的健身计划，评估其效果。这样的多元化应用不胜枚举，涉及人们生活的方方面面，满足不同人群的各种需求。

公有云平台的出现推动了互联网应用的创新和发展。这些平台降低了云应用的开发、运营、维护成本。从基础设施到必备软件，从应用的可伸缩性到运行时的服务质量保障，这一切都将由云平台来处理。那么，对于云应用提供商，尤其是多元应用提供商来说，一款云应用的诞生甚至可以实现零初始投入的目标，唯一需要的就是富有创意的点子和敏捷而简单的开发。

上面我们将云应用划分为三种类型，这三种类型的划分可以使用“长尾理论”来诠释。在如图6.9所示的长尾模型中，横轴是云应用的种类，纵轴是云应用的流行程度。少量的标准应用具有最高的流行度，成为长尾图形的“头”。中等规模的客户应用具有中等的流行度，成为长尾图形的

“肩”。大量的多元应用具有较低的流行度，成为长尾图形的“尾”。

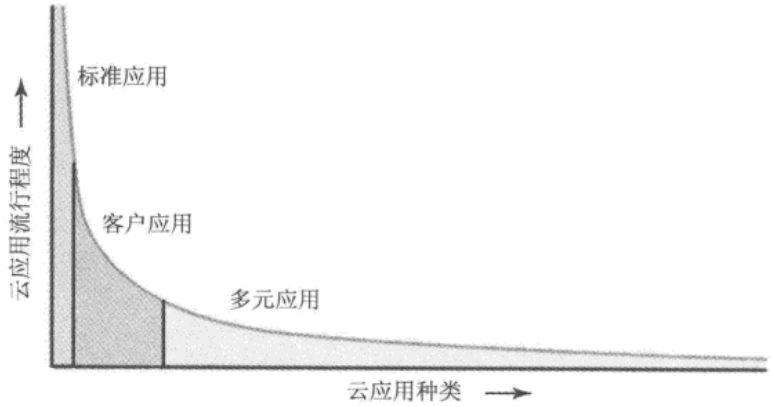



图6.9　云应用的长尾模型

标准应用是人们日常生活中不可或缺的服务，比如文档处理、电子邮件和日程管理等。这些应用提供的功能是人们所熟悉的，绝大多数云应用使用者将会使用它们来处理日常事务。标准应用的类型有限，它们必须具备的功能和与用户交互的方式在一定程度上已经形成了业界标准。标准应用的提供商往往是具有雄厚实力的IT业巨头。

客户应用针对的是某种具有普遍性的需求，比如客户管理系统（CRM）和企业资源计划系统（ERP）等。这样的应用可以被不同的客户定制，为数量较大的用户群所使用。客户应用的类型较丰富，但往往集中在若干种通用的业务需求上。客户应用的提供商可以是规模较小的专业公司。

多元应用满足的往往是小部分用户群体的个性化需求，比如身处某个城市的居民或者正在进行健身练习的用户。这样的应用追求新颖和快速，虽然应用的用户群体可能有限，但是它却对该目标群体有着巨大的价值。多元应用的种类繁多，千变万化，其提供者可以是规模很小的开发团队，甚至是个人。

“长尾理论”的核心思想是：再微小的需求如果能够得到满足，就可以创造价值。而这些微小需求的集合就是长尾的尾，它聚合起来具有巨大的潜力。在云应用的生态系统中，客户应用和多元应用落在长尾的肩部和尾部。在传统信息产业模式中，这部分空间所蕴藏的价值并没有被很好地挖掘。各大IT厂商主要关注于长尾的头部，而忽视了相对较难把握的个性化需求。云计算的出现显著降低了应用的开发和维护成本，拉近了初创型公司和行业巨头们的技术差距，使得具有创新精神和独到眼光的团队可

以快速地将构想化为现实。可以说，云计算为信息行业创造了新的增长空间，也为互联网用户提供了更加丰富的选择。

### 1. 标准应用

在线文档服务是标准应用的一个典型示例，比如Google Docs。Google Docs允许用户在线创建文档，并提供了多种布局模板。Google Docs是完全基于浏览器的SaaS服务，用户不需在本地安装任何程序，只需要通过浏览器登录服务器，就可以随时随地获得自己的工作环境。在用户体验上，该服务做到了尽量符合用户使用习惯，不论是页面布局、按钮菜单设置还是操作方法都与用户所习惯的本地文档处理软件（如Microsoft Office和OpenOffice等）相似。用户可以从零开始采用该标准应用创建新文档，也可以将现有文档上传到应用服务器端，利用Google Docs的处理功能继续编辑。编辑工作完成后，用户可以将其下载到本地机器保存，也可以将其保存在服务器端。将文档保存在服务器端的好处是可以方便地利用该标准应用提供的共享功能与预先设定的合作者共同创作文档，或者邀请审阅者对文档进行在线审阅。Google Docs还支持将编辑好的文档发布到互联网，用户可以设定访问权限，让全世界的互联网用户或者一部分指定的用户像浏览网页一样看到发布出来的文档。

标准应用的一个重要特点就是代码运行在平台层上，而不是用户本地的机器上。很多以前在本地运行的复杂应用将陆续被迁移到云中，并且由用户通过浏览器来执行。这就需要在网页中提供和本地窗口应用一样丰富的功能集合，并且在服务质量（比如响应速度）上和本地窗口应用差别不大。然而，这类云应用在功能方面往往与先前本地的版本有所差异，这很大程度上是因为云应用的开发难度要大很多。先前本地版本的应用有着经过几十年不断改进过的编程语言和大量开发工具的支持，而在线应用的开发则主要依赖于JavaScript，在开发和调试的难度上都比较高，而且需要额外考虑远程通信的效率问题。如果能够基于目前比较主流的编程语言开发应用，然后在运行时生成优化的JavaScript代码，则可以在很大程度上简化开发的复杂度，Google Web Toolkit正是朝着这一方向的一个尝试，使得开发人员可以使用Java语言开发支持Ajax的Web应用。

### 2. 客户应用

Salesforce CRM是客户应用的典型代表。其关键点在于采用了多租户

架构，使得所有用户和应用程序共享一个实例，同时又能够按需满足不同的客户要求。多租户架构分离了应用的逻辑和数据，企业用户可以通过元数据定义自己的行为和属性，并且定制化以后的应用程序不会影响其他企业用户。另外，Salesforce.com还推出了自己的编程语言Apex，它是一个易用的、多租户的编程语言，在一定程度上解决了应用层在模型开发复杂度方面的问题。用户可以通过Apex创建自己的组件，修改Salesforce.com提供的现有代码。不仅如此，Apex还使得编写的程序天生就符合网络服务的要求，并且可以通过SOAP方式访问，方便了第三方的ISV进行应用开发。

在开发结束以后，应用能够被有效地部署在运行平台上，并激活至可用状态。对于用户来说，应用达到可用状态并不是唯一的目标，应用还需要具有一定的互操作性。互操作性一方面是考虑如何将现有的应用迁移到云中，另一方面是考虑云应用是否可以从一个云提供商迁移到另外一个不同的云提供商。前者的问题其实和传统意义上的互操作性比较类似，考虑的是应用从一个操作系统迁移到另外一个操作系统，或者将应用从一个运行平台迁移到另外一个运行平台。后者的问题主要是由于目前云计算缺乏一整套开放标准，使得云应用乃至整个云计算自身缺乏统一的数据描述模型及通信标准等规范。如果云应用不能迁移，那么当用户决定选择另外一个云提供商作为其服务平台的时候，就意味着他先前的投入没有被有效地再利用。更为致命的是用户的数据将无法从一个云提供商中导出并导入到另一个云提供商。这无疑会使用户，尤其是拥有大量历史数据的企业和机构用户对云计算望而却步，这对于云计算本身的发展是极为不利的。互操作性的解决有赖于建立云计算的开放标准，这需要当前IT公司的共同推动。





### 3. 多元应用

为旧金山地区用户提供实时、随处的公交系统时刻表服务的Mutiny是多元应用的典型代表之一。用户可以随时通过便携设备登录Mutiny网站，获知自己所处位置附近所有的公共汽车、地铁线路和停靠站点，以及下一班车的进站时间。Mutiny获取移动设备上的GPS坐标，利用该坐标信息访问Google Map的API得到使用者目前所处的街道位置，以及其附近所有的公交站、地铁站信息。用户单击其中任意一个站点，就会得到这个站点下一班车的到站时间，该到站信息是从旧金山市公共交通系统的网站上获得的。可见，Mutiny巧妙地整合网络上的数据资源，利用云平台为特定用户

群（旧金山市的居民）提供了便捷的服务。

以Mutiny为代表的云应用通常将来自两个或多个源的数据进行组合，构成一个崭新的服务。这种设计方式被称为Mashup，它追求的是便捷而快速的整合，通常是使用数据源提供的开放应用程序接口（Open API）来实现的。Mashup应用在架构上由两个不同部分组成：数据内容/Open API提供者和Mashup站点。这两个部分在逻辑上和物理上都是相互分离的。数据内容/Open API提供者是被融合的内容的提供者。在Mutiny的例子中，该提供者是Google Map和旧金山市的公交系统网站。为了方便数据的检索，数据源通常会将自己的内容通过Web协议对外提供。Mashup站点是数据融合发生的地方，可以在服务器端完成，也可以在浏览器端完成。若在服务器端，Mashup直接使用服务器端动态内容生成技术实现，为用户提供整合后的最终页面；若在浏览器端，则需通过客户机端脚本（如JavaScript）或Applet来完成。

## 6.5 小结

本章介绍了云架构的基本层次，即基础设施层、平台层和应用层。对基础设施层和平台层我们都介绍了定义、基本功能和参考示例；对应用层我们介绍了其分类和示例。本章要点如下。

（1）基础设施层是物理硬件和一系列软件的集合，它的主要功能是抽象物理硬件资源，在基础设施层内部实现自动化的资源管理和优化，并为外部使用者提供各种各样的基础设施层服务，使得硬件资源可以很容易地被访问和管理。基础设施层的基本功能包括资源抽象、资源监控、资源部署、负载管理、数据管理、安全管理和计费管理。

（2）平台层是指建立在基础设施层上的一系列软件，平台本身提供的软件服务可以在应用的开发过程中被快速集成，从而显著缩短应用的开发周期；开发人员通过互联网将应用远程发布至平台层而不需要关心应用被发布在哪台服务器上，即平台层的物理信息对于使用者是透明的；云应用的运行不受资源限制，并且以一种按量计费的模型对云应用收费。平台层的基本功能是提供开发平台、运行平台、管理平台和监控计费。

（3）云应用是指运行在平台层之上、以软件即服务的形式提供给客户的应用。应用层的分类主要包括标准应用、客户应用和多元应用三类。

CHAPTER 7

# 第7章 云计算的关键技术与挑战

作为信息产业在互联网时代的最新发展，云计算在过去的几年中快速成长，已经出现了很多商业应用，比如Amazon EC2、Google App Engine和Salesforce.com提供的不同层次的云服务。云计算的发展和应用离不开一系列创新技术的支持，云计算所追求的愿景也为当下互联网时代的信息技术带来了一系列挑战。在这些挑战中，有些虽然是传统计算平台中的经典问题，但在云计算中又被赋予了新的内涵；有的则是云计算中的新问题、新技术。本章将介绍云计算关键技术产生的背景、要解决的问题和发展现状等，并分析云计算环境下的技术挑战。

## 7.1 云计算的关键技术

云计算的理念生动体现了互联网时代的信息服务特性，并且正在推动一系列技术创新去解决互联网平台的服务生命周期管理问题，大规模分布式计算、存储、通信问题，以及资源按需提供、按量计费问题。本节将从技术维度更深入地讨论云计算，着重介绍云计算中的快速部署、资源调度、多租户、海量数据处理、大规模消息通信、大规模分布式存储、许可证管理与计费等关键技术。

### 7.1.1 快速部署

自数据中心诞生以来，快速部署就是一项重要的功能需求。数据中心管理员和用户一直在追求更快、更高效、更灵活、功能更齐全的部署方案。云计算环境对快速部署的要求将会更高。首先，在云环境中资源和应用不但规模变化范围大而且动态性高。用户所需的服务主要采用按需部署的

方式，即用户随时提交对资源和应用的请求，云环境管理程序负责分配资源、部署服务。其次，不同层次云计算环境中服务的部署模式是不一样的，比如虚拟化的基础设施云上的应用都被打包在虚拟机里面，而多租户平台上的应用则会选择更加轻量级的打包方案。另外，部署过程所支持的软件系统形式多样，系统结构各不相同，部署工具应能适应被部署对象的变化。

在第3章中我们介绍了基于流传输的虚拟机部署方法，该方法可以有效减少单个虚拟机的部署时间，但是包含了操作系统、中间件、应用软件的虚拟机镜像，大小通常为几个GB到几十个GB，镜像的复制速度会严重影响虚拟机的部署速度和用户体验。另外，虚拟机的激活涉及到整个软件栈的配置和关联关系，操作非常复杂，自动化程度的高低直接关系着虚拟机部署的效率。因此，即使采用了流传输来部署，这个过程仍然会耗费大量时间。此外，在部署多个虚拟机时，基于流传输的虚拟机部署采用的是顺序的、串行的部署方法，如果想进一步提高云环境中虚拟机的部署速度，则需要考虑并行部署或者协同部署技术。

并行部署是指将传统的顺序部署方式改变为并行执行，同时执行多个部署任务，将虚拟机同时部署到多个物理机上，如图7.1所示。

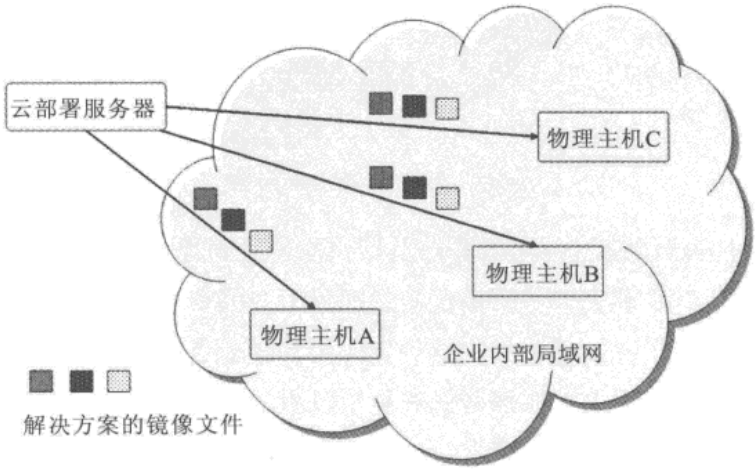



图7.1　并行部署系统架构

理想情况下，并行部署可以成倍地减少部署所需时间，但存储镜像文件所在的部署服务器的读写能力或者部署系统的有限网络带宽却制约实际的并行程度即部署速度。例如，在网络带宽有限的情况下，同时运行多个部署任务时，这些任务会争抢网络带宽，当网络带宽被占满时，部署速度就不能进一步提高了。在这种情况下，协同部署技术可以用来进一步提高部署速度。

协同部署技术的核心思想是将虚拟机镜像在多个目标物理机之间的网

络中传输，而不是仅仅在部署服务器和目标物理机之间传输，从而提高部署速度。通过协同部署，部署服务器的网络带宽不再成为制约部署速度的瓶颈，部署的速度上限取决于目标物理机之间的网络带宽的总和。基于虚拟化技术和协同部署技术，我们可以构建一个协同部署系统，从而保证大规模数据中心中服务的部署速度、效率和质量。如图7.2所示，协同部署系统的架构包括了部署服务器节点（图中的云部署服务器）和被部署节点（图中的物理主机A、B、C），关键模块包括部署控制器、镜像拷贝器、协同部署器和协同控制器等。

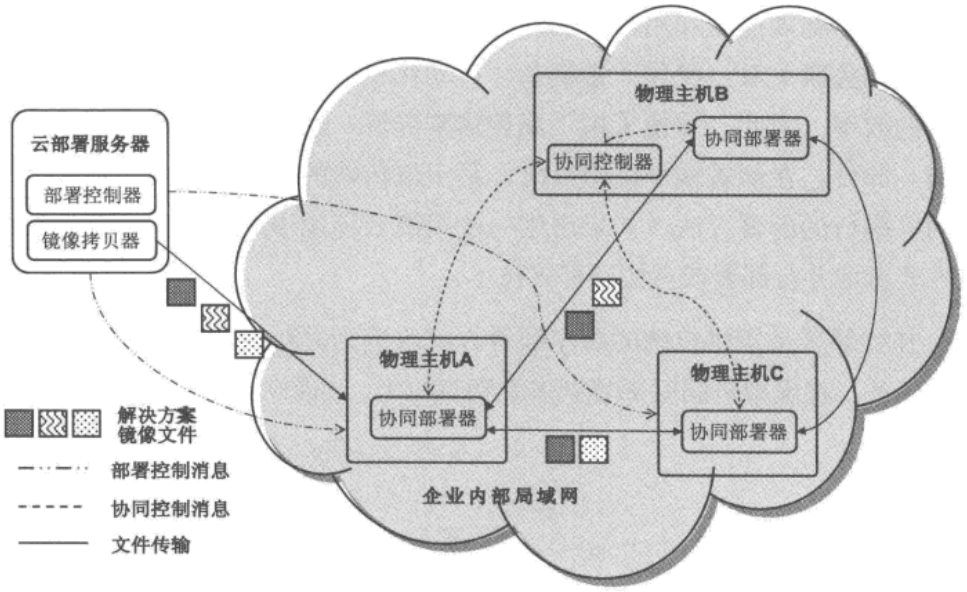



图7.2　协同部署系统架构





部署服务器负责将协同部署器及用户空间文件系统（通过I/O操作截获技术，将用户的本地文件访问重定向到网络上）的安装文件发送到被部署节点，并发起部署任务；部署控制器负责协调各个节点之间的部署进度，交换文件片信息；被部署节点在部署任务开始以后，根据启动顺序向用户空间文件系统发出虚拟镜像文件块请求，用户空间文件系统调用协同部署器获取文件块。协同部署技术能够大大提高系统部署的速度。由于物理机之间存在大量的共享带宽，因此协同部署可能会影响其他物理机的网络带宽。

并行部署和协同部署技术同样可以运用到物理解决方案的自动化部署过程中，加速部署过程。云环境中物理解决方案的部署是指在物理平台上安装软件环境。首先，在云的硬件环境搭建起来以后，需要在这些硬件上安装云的软件环境，这就涉及到大规模的操作系统的部署、虚拟机运行平台的配置、云基础设施层管理软件的安装等。其次，在扩展云平台架构的

时候（例如为现有的数据中心加入新的物理机），需要在新节点上面部署和配置操作系统、虚拟化平台、中间件等全套软件。

与虚拟机的部署相比，物理解决方案自动化部署的难点在于软件的多样性和解决方案的复杂性。为了能够自动化部署物理解决方案，需要定义一种标准的解决方案打包格式，将软件程序文件、安装配置脚本、元数据等内容一起打包；还需要一个通用的部署引擎和一组自动化安装配置流程。通过这种方式，部署引擎在接收到解决方案的打包文件以后，能够解析解决方案的元数据，按照自动化流程驱动整个解决方案的安装配置过程。

## 7.1.2 资源调度

资源调度指的是在特定的资源环境下，根据一定的资源使用规则，在不同的资源使用者之间进行资源调整的过程。这些资源使用者对应着不同的计算任务（例如一个虚拟化解决方案），每个计算任务在操作系统中对应于一个或者多个进程。通常有两种途径可以实现计算任务的资源调度：在计算任务所在的机器上调整它的资源使用量，或者将计算任务转移到其他机器上。图7.3是将计算任务迁移到其他机器上的一个例子。在这个例子中，物理资源A（如一台物理服务器）的使用率远高于物理资源B，通过将计算任务1从物理资源A迁移到物理资源B，使得资源的使用更加均衡和合理，从而达到负载均衡的目的。

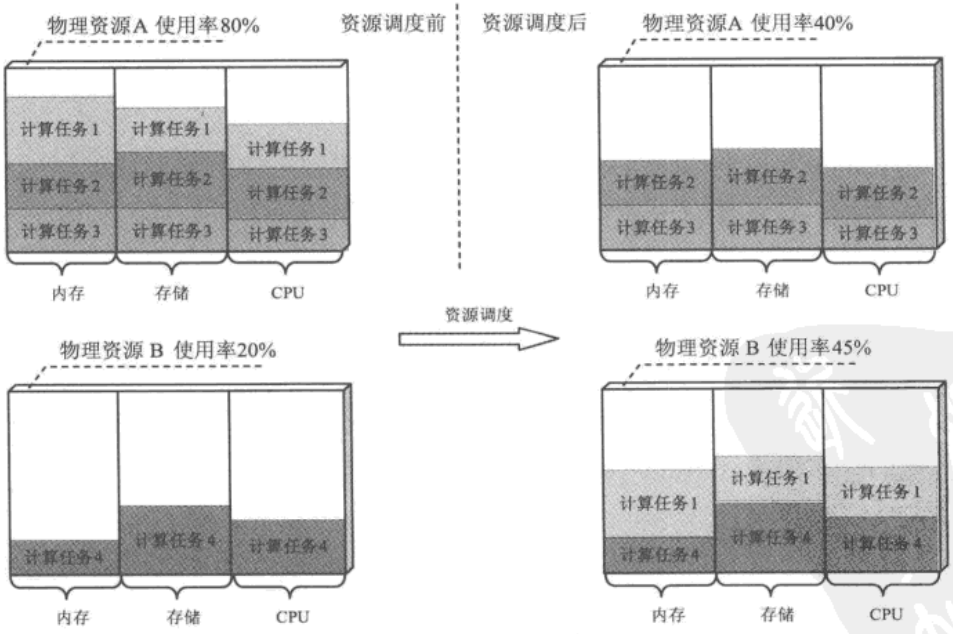



图7.3 资源调度

目前的技术已经实现了在几秒钟内（暂时停机时间为毫秒级）将一个操作系统进程从一台机器迁移到另一台机器。这种操作系统进程的动态迁移技术能够实现计算任务在不同的机器之间的迁移。虚拟机的出现使得所有的计算任务都被封装在一个虚拟机内部。由于虚拟机具有隔离特性，因此可以采用虚拟机的动态迁移方案来达到计算任务迁移的目的。

云计算的海量规模为资源调度带来了新的挑战。资源调度需要考虑到资源的实时使用情况，这就要求对云计算环境的资源进行实时监控和管理。云计算环境中资源的种类多、规模大，对资源的实时监控和管理就变得十分困难。此外，一个云计算环境可能有成千上万的计算任务，这对调度算法的复杂性和有效性提出了挑战。对于基于虚拟化技术的云基础设施层，虚拟机的大小一般都在几个GB以上，大规模并行的虚拟机迁移操作很有可能会因为网络带宽等各因素的限制而变得非常缓慢。

从调度的粒度来看，虚拟机内部应用的调度是云计算用户更加关心的。如何调度资源满足虚拟机内部应用的服务级别协定也是目前待解的一个难题。以性能为例，一个应用资源调度系统需要监控应用的实时性能指标，例如吞吐量、响应时间等。通过这些性能指标，结合历史记录及预测模型，分析出未来可能的性能值，并与用户预先制定的优化规则进行比较，得出应用是否需要及如何进行资源调整的结论。目前，大多数虚拟化管理方案只能通过在虚拟机级别上的调度技术结合一定的调度策略来尝试为虚拟机内部应用做资源调度，普遍缺乏精确性和有效性。





### 7.1.3 多租户技术

传统的软件运行和维护模式要求软件被部署在用户所购买或租用的数据中心当中，这些软件大多服务于特定的个人用户或者企业用户。在云计算环境中，更多的软件以SaaS的方式发布出去，并且通常会提供给成千上万的企业用户共享使用。与传统的软件运行和维护模式相比，云计算要求硬件资源和软件资源能够更好地共享，具有良好的可伸缩性，任何一个企业用户都能够按照自己的需求对SaaS软件进行客户化配置而不影响其他用户的使用。多租户（Multi-Tenant）技术就是目前云计算环境中能够满足上述需求的关键技术。

多租户技术是一项云计算平台技术，该技术使得大量的用户能够共享同一堆栈的软、硬件资源，每个用户能够按需使用资源，能够对软件服务

进行客户化配置，而且不影响其他用户的使用。这里，每一个用户被称为一个租户，如图7.4所示。

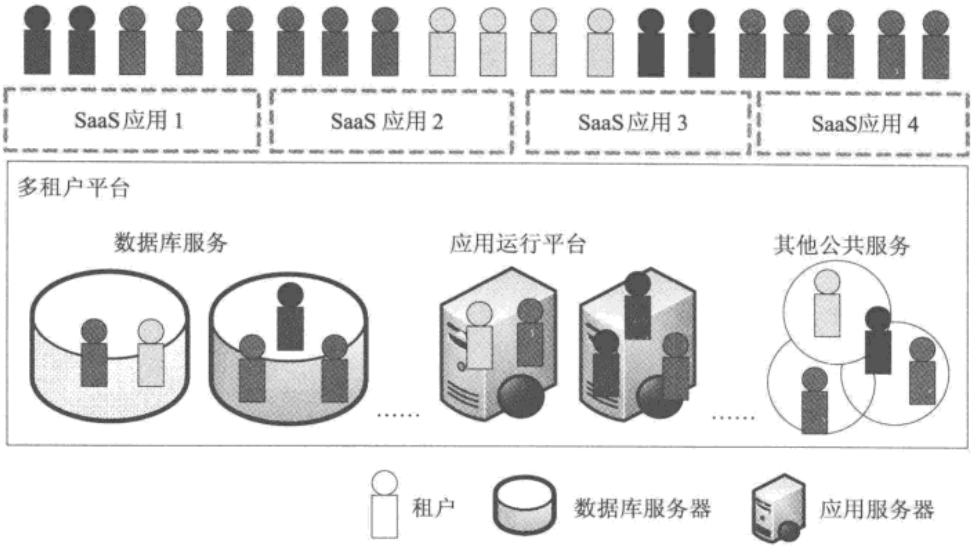



图7.4　多租户平台

目前普遍认为，采用多租户技术的SaaS应用需要具有两项基本特征：第一点是SaaS应用是基于Web的，能够服务于大量的租户并且可以非常容易地伸缩；第二点则在第一点的基础上要求SaaS平台提供附加的业务逻辑使得租户能够对SaaS平台本身进行扩展，从而满足更大型企业的需求。目前，多租户技术面临的技术难点包括数据隔离、客户化配置、架构扩展和性能定制。

数据隔离是指多个租户在使用同一个系统时，租户的业务数据是相互隔离存储的，不同租户的业务数据处理不会相互干扰。多租户技术需要实现安全、高效的数据隔离，从而保证租户数据安全及多租户平台的整体性能。对多租户的数据库管理有三种基本方式：第一种方式是给每一个租户创建单独的数据库，这样做的好处是用户间数据充分隔离，缺点是数据库管理的成本和开销比较大；第二种方式是将多个租户的数据保存在同一个数据库中，但是采用不同的Schema，这样在一定程度上减少了数据库的管理成本和开销，但是相应地影响了数据隔离的效果；第三种方式是将多个租户的数据保存在一个数据库中，采用相同的Schema，也就是说将数据保存在一个表或者一类具有相同Schema的表中，通过租户的标识码字段进行区别，这样的管理成本和开销最低，但是数据隔离的效果最差，需要大量的安全性检验来保障租户间的数据隔离。

客户化配置是指SaaS应用能够支持不同租户对SaaS应用的配置进行

定制，比如界面显示风格的定制等。客户化配置的基本要求是一个租户的客户化操作不会影响到其他租户。这就要求多租户系统能够对同一个SaaS应用实例的不同租户的配置进行描述和存储，并且能够在租户登录SaaS应用时根据该租户的客户化配置为其呈现相应的SaaS应用。在传统的企业应用运行模式中，每个企业用户都拥有一个独立的应用实例，因此可以非常容易地存储和加载任何客户化配置。但在多租户场景下，成千上万的租户共享同一个应用实例。在现有的平台技术中，比如J2EE，对应用配置的更改通常会对该平台中的所有用户产生影响。因此，如何支持不同租户对同一应用实例的独立客户化配置是多租户技术面临的一个基本挑战。

架构扩展是指多租户服务能够提供灵活的、具备高可伸缩性的基础架构，从而保证在不同负载下多租户平台的性能。在典型的多租户场景中，多租户平台需要支持大规模租户的同时访问，因此平台的可伸缩性至关重要。一个最简单的方法是在初始阶段就为多租户平台分配海量的资源，这些资源足以保证在负载达到峰值时的平台性能。然而，很多时候负载并不是处于峰值的，所以这个方法会造成巨大的计算资源和能源浪费，并且会大幅增加多租户平台提供商的运营成本。因而，多租户平台应该具有灵活可伸缩的基础架构，能够根据负载的变化按需伸缩。

性能定制是多租户技术面临的另一个挑战。对于同一个SaaS应用实例来说，不同的用户对性能的要求可能是不同的，比如某些客户希望通过支付更多的费用来获取更好的性能，而另一些客户则本着“够用即可”的原则。在传统的软件运营模式中，由于每个客户拥有独立的资源堆栈，只需要简单地为付费多的用户配置更高级的资源就可以了，因此相对而言性能定制更容易一些。然而，同一个SaaS应用的不同租户共享的是同一套资源，如何为不同租户在这一套共享的资源上灵活地配置性能是多租户技术中的难点。

IT人员经常会面临选择虚拟化技术还是多租户技术的问题。多租户与虚拟化的不同在于：虚拟化后的每个应用或者服务单独地存在一个虚拟机里，不同虚拟机之间实现了逻辑的隔离，一个虚拟机感知不到其他虚拟机；而多租户环境中的多个应用其实运行在同一个逻辑环境下，需要通过其他手段，比如应用或者服务本身的特殊设计，来保证多个用户之间的隔离。多租户技术也具有虚拟化技术的一部分好处，如可以简化管理、提高服务器使用率、节省开支等。从技术实现难度的角度来说，虚拟化已经比

较成熟，并且得到了大量厂商的支持，而多租户技术还在发展阶段，不同厂商对多租户技术的定义和实现还有很多分歧。当然，多租户技术有其存在的必然性及应用场景。在面对大量用户使用同一类型应用时，如果把每一个用户的应用都做成单独的虚拟机，可能需要成千上万台虚拟机，这样会占用大量的资源，而且有大量重复的部分，虚拟机的管理难度及性能开销也大大增加。在这种场景下，多租户技术作为一种相对经济的技术就有了用武之地。

### 7.1.4 海量数据处理

作为以互联网为计算平台的云计算，将会更广泛地涉及到海量数据处理任务。海量数据处理指的是对大规模数据的计算和分析，通常数据的规模可以达到TB甚至PB级别。在互联网时代，互联网数据的统计和分析很多是海量数据级别的，一个典型的例子就是搜索引擎。由于数据量非常大，一台计算机不可能满足海量数据处理的性能和可靠性等方面的要求。以往对于海量数据处理的研究通常是基于某种并行计算模型和计算机集群系统的。并行计算模型可以支持高吞吐量的分布式批处理计算任务和海量数据，计算机集群系统则在通过互联网连接的机器集群上建立一个可扩展的可靠的计算环境。

在互联网时代，由于海量数据处理操作非常频繁，很多研究者在从事支持海量数据处理的编程模型方面的研究。例如，Remzi等人在1999年设计了River编程模型，开发人员可以基于该编程模型进行开发和执行计算任务。River编程模型的设计目的就是使得基于大规模计算机集群的编程和计算更加容易，并且获得极佳的计算性能。River编程模型有两个核心设计特性：一个高性能的分布式队列和一个存储冗余机制。因此，River需要对磁盘和网络的数据传输进行非常精心的调度。当今世界最流行的海量数据处理的编程模型可以说是由Google公司的Jeffrey Dean等人所设计的MapReduce编程模型。MapReduce编程模型将一个任务分成很多更细粒度的子任务，这些子任务能够在空闲的处理节点之间调度，使得处理速度越快的节点处理越多的任务，从而避免处理速度慢的节点延长整个任务的完成时间。下面我们将介绍MapReduce框架的工作原理和设计原则，从而加深读者对海量数据处理系统的理解。

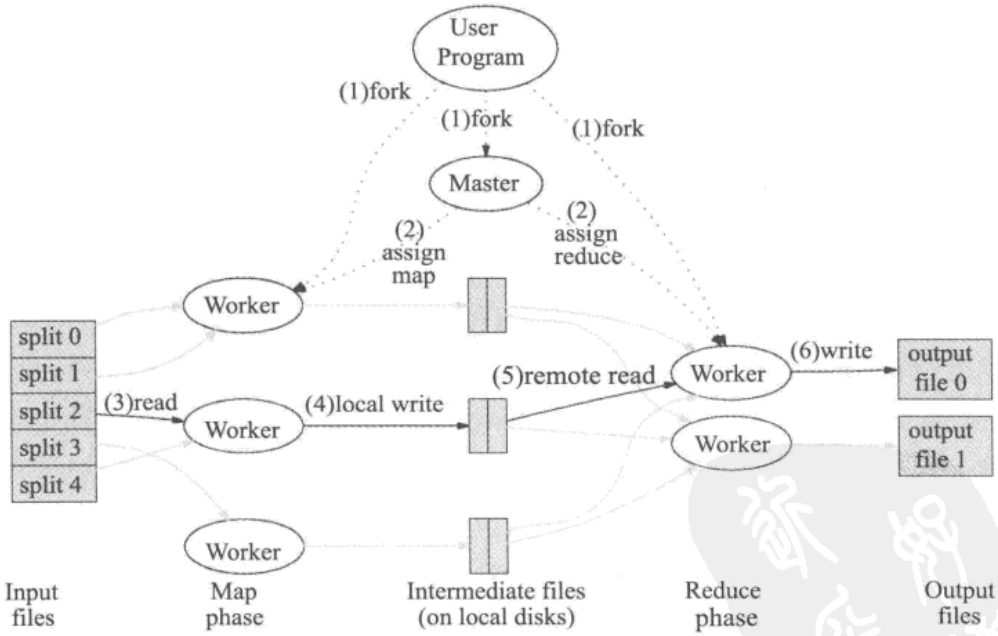
MapReduce框架从Lisp及很多其他类似的语言获得灵感，研究人员发

现大多数分布式运算可以抽象为Map和Reduce两个步骤，从而实现可靠、高效的分布式应用。Map步骤负责根据输入的键值（key/value）对生成中间结果，中间结果同样采用key/value对的形式。Reduce步骤则将所有的中间结果根据key进行合并，然后生成最终结果。开发者只需要实现Map和Reduce函数的逻辑，然后提交给MapReduce运行环境，计算任务便会在由大量计算机组成的集群上被自动、并行地调度执行。运行环境负责将输入数据进行分割、调度任务、自动处理运行过程中的机器失效，以及协调不同节点之间的数据通信。图7.5描绘了Jeffrey Dean等人所设计的MapReduce框架的基本工作流程。

MapReduce的运行环境由两种不同类型的节点组成：Master和Worker。Worker负责数据处理，Master负责任务调度及不同节点之间的数据共享。具体执行流程如下。

（1）利用MapReduce提供的库将输入数据切分为M份，每份的大小为16～64MB，然后在计算机集群上启动程序。

（2）Master节点的程序负责为所有Worker节点分配子任务，其中包括M个Map子任务和R个Reduce子任务。Master负责找出空闲的节点并分配子任务。




图表来源：MapReduce论文

图7.5　MapReduce框架的基本工作流程

（3）获得Map子任务的Worker节点读入对应的输入数据，从输入数据中解析key/value对，并调用用户编写的Map函数。Map函数的中间结果缓存在内存中并周期性地写入本地磁盘。写入本地磁盘的数据根据用户指定的划分函数被分为R个数据区。这些中间结果的位置被发送给Master节点。Master节点继续将这些数据信息发给负责Reduce任务的Worker节点进行Reduce处理。

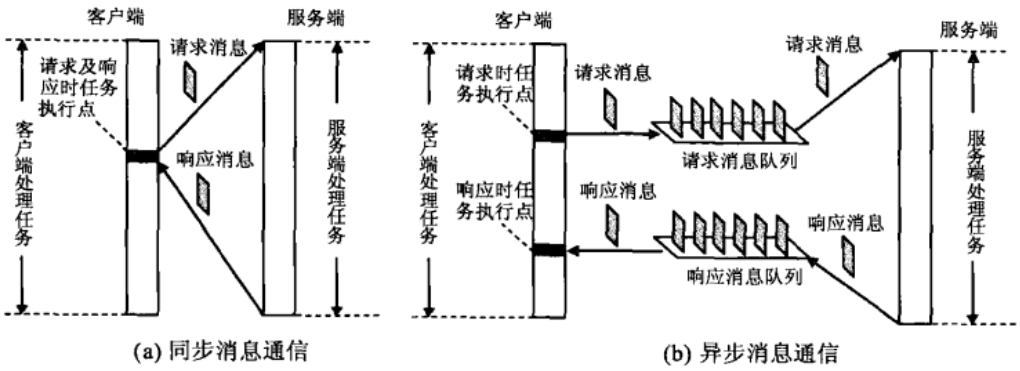
（4）执行Reduce子任务的Worker节点从Master节点获取子任务后，使用远程调用的方式从执行Map任务的Worker节点的本地磁盘读取数据到缓存。执行Reduce子任务的Worker节点首先遍历所有的中间结果，然后按照关键字进行排序。

（5）执行Reduce子任务的Worker节点遍历获得Map子任务产生的中间数据，将每个不同的key和value进行结合并传递给用户的Reduce函数。Reduce函数的结果被写入到一个最终的输出文件。当所有的Map子任务和Reduce子任务完成的时候，Master节点将R份Reduce结果返回给用户程序。用户程序可以将这些执行Reduce子任务的Worker节点生成的结果数据合并得到最终结果。

在设计MapReduce的时候，研究人员考虑了很多大规模分布式计算机集群进行海量数据处理时所要考虑的关键问题：容错处理保证了在Master和Worker都失效的情况下计算任务仍然能够正确执行；操作本地化保证了在网络等资源有限的情况下，最大程度地将计算任务在本地执行；任务划分的粒度使得任务能够更加优化地被分解和并行执行；对于每个未完成的子任务，Master节点都会启动一个备份子任务同时执行，无论初始任务还是备份子任务处理完成，该子任务都会立即被标记为完成状态，通过备份任务机制可以有效避免因个别节点处理速度过慢而延误整个任务的处理速度。

### 7.1.5 大规模消息通信

云计算的一个核心理念就是资源和软件功能都是以服务的形式进行发布的，不同服务之间经常需要通过消息通信进行协作。可靠、安全、高性能的通信基础设施对于云计算的成功至关重要。通常，消息通信可以分为同步通信和异步通信两种方式，如图7.6所示。




(a) 同步消息通信　　(b) 异步消息通信

图7.6　同步通信和异步通信

在同步消息通信中，直接请求服务器端的服务，并等待服务结果返回后才继续执行；在服务端，服务的运行环境则需要保存与客户端通信的信息，在处理完成时将结果返回给客户端。这种同步消息通信机制有可能对客户端系统的处理速度和服务端系统的可用性造成影响：首先，客户端系统因为需要同步等待而无法继续处理任务；其次，同步通信机制造成服务端系统资源长时间被占用，服务实例也由于需要与远程客户端通信而无法在任务处理完成时立即处理下一个任务；另外，同步消息通信会降低服务的可用性，因为在分布式环境中，客户端所请求的服务实例有可能因为各种原因而不可用，从而造成客户端请求无法得到处理。因此，异步消息通信对于云计算环境就显得尤为重要。

在异步消息通信中，客户端和服务端并不直接通信。客户端把请求以消息的形式放在请求消息队列里面，然后继续处理其他业务逻辑；服务实例则会从请求消息队列中获取请求消息，并且将处理结果放入响应消息队列里面，然后立即处理下一个请求。消息通信管理软件通过判断消息请求是否成功发给目标服务实例来判断该实例是否可用，并且在目标服务实例不可用的情况下将消息发给其他服务实例，从而为客户端提供高可用的服务。

异步消息通信机制已经经过了多年的发展。早在1995年就提出了基于生产者/消费者模型的分布式消息队列方案，并且能够根据分析模型考量和预测消息队列的性能。Java Message Service（JMS）是J2EE平台上的一个消息通信标准，J2EE应用程序可以通过JMS来创建、发送、接收和阅读消息。Apache ActiveMQ是JMS的一个开源实现版本，IBM WebSphere MQ则是实现了JMS的一个商业产品，并且通过一系列的增强特性提高了JMS消息通信的性能和可管理性。异步消息通信已经成为面向服务架构中组件解耦合及业务集成的重要技术。

面向服务的理念使得异步消息通信对云计算更加重要。异步消息通信机制可以使得云计算每个层次中的内部组件之间及各个层次之间解耦合，并且保证云计算服务的高可用性。异步消息通信机制对于服务的可伸缩性也非常重要，消息队列管理软件可以通过队列中的消息数量和消息请求的服务类型预测每种服务的工作负载变化趋势，并且通过该趋势自动增加和减少服务实例。

云计算也给分布式系统中的消息通信带来了新的挑战。首先，消息通信服务必须足够稳定，以保证在应用程序需要使用消息服务的时候该服务一定是可用的，并且保证消息在互联网传输过程中不会丢失。其次，消息通信服务必须能够伸缩，从而支持大规模节点同时执行高性能的消息读写操作。云计算的安全问题一直以来备受关注，因此消息通信服务还要保证消息的传递是安全的，从而保证业务是安全的。此外，紧凑、高效的消息内容模型对提高消息处理效率非常重要，这在云计算这样的大规模消息通信处理环境中体现得尤为明显。目前，云计算环境中的大规模数据通信技术仍在发展阶段，Amazon公司的Simple Queue Service（SQS）是当今业界著名的云计算大规模消息通信产品，在第8章将对该产品进行介绍。

### 7.1.6 大规模分布式存储

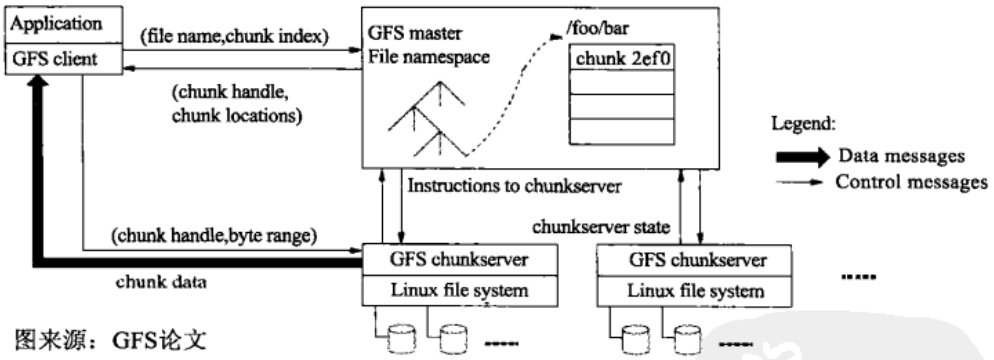
分布式存储的目标是利用多台服务器的存储资源来满足单台服务器所不能满足的存储需求。分布式存储要求存储资源能够被抽象表示和统一管理，并且能够保证数据读写操作的安全性、可靠性、性能等各方面要求。

随着过去几十年互联网技术的发展，越来越多的互联网应用具有存储海量数据的需求，比如搜索引擎和互联网视频网站，这些需求催生了一些优秀的大规模分布式存储技术，比如分布式文件系统。分布式文件系统允许用户像访问本地文件系统一样访问远程服务器的文件系统，用户可以将自己的数据存储在多个远程服务器上，分布式文件系统基本上都有冗余备份机制和容错机制来保证数据读写的正确性。云环境的存储服务基于分布式文件系统并根据云存储的特征做了相应的配置和改进。下面分别介绍分布式文件系统和云存储服务。

下面介绍几个典型的分布式文件系统。Frangipani是一个可伸缩性很好的高性能分布式文件系统，该系统采用了两层的服务体系架构：底层是一个分布式存储服务，该服务能够自动管理可伸缩、高可用的虚拟磁盘；在这

个分布式存储服务上层运行着Frangipani分布式文件系统。JetFile是一个基于P2P的组播技术、支持在Internet这样的异构环境中分享文件的分布式文件系统。Ceph是一个高性能并且可靠的分布式文件系统，它通过把数据和对数据的管理在最大程度上分开来获取极佳的I/O性能。

Google File System（GFS）是Google公司设计的可伸缩的分布式文件系统。Google公司的工程师在考虑了分布式文件系统的设计准则的基础上，又发现了以下几个不同于传统分布式文件系统的需求：第一，PC服务器极易发生故障，造成节点失效，故障的原因多种多样，有机器本身的、网络的、管理员引起的及外部环境引起的，因此需要对整个系统中的节点进行监控，检测出现的错误，并开发相应的容错和故障恢复机制；第二，在云计算环境中，海量的结构化数据被保存为非常大的文件，一般为GB量级，因此需要改变原有的基于对中小文件（KB或者MB量级）进行管理的文件系统设计准则，以适应对超大文件的访问；第三，系统中对文件的写操作绝大多数是追加操作，也就是在文件的末尾写入数据，在文件中间写入数据的情况其实很少发生，而且数据一旦被写入，绝大多数情况下都是被顺序地读取，不会被修改，因此在设计系统时把优化重点放在追加操作上，就可以大幅度提高系统的性能；第四，设计系统时要考虑开放的、标准的操作接口，并隐藏文件系统下层的负载均衡、冗余复制等细节，这样才可以方便地被上层系统大量地使用。因此，GFS能够很好地支持大规模海量数据处理应用程序。图7.7展示了GFS的系统架构。




图来源：GFS论文

图7.7　Google File System架构图

云计算的出现给分布式存储带来了新的需求和挑战。在云计算环境中，数据的存储和操作都是以服务的形式提供的；数据的类型多种多样，包括了普通文件、虚拟机镜像文件这样的二进制大文件、类似XML的格式化数据，甚至数据库的关系型数据等；云计算的分布式存储服务设计必须

考虑到各种不同数据类型的大规模存储机制，以及数据操作的性能、可靠性、安全性和简单性。

目前，在云计算环境下的大规模分布式存储方向已经有了一些研究成果和应用。BigTable是Google公司设计的用来存储海量结构化数据的分布式存储系统，Google公司使用该系统来将网页存储成分布式的、多维的、有序的图。Dynamo是Amazon公司设计的一种基于键值对的分布式存储系统，该系统在设计之初的一个主要考虑就是Amazon公司的大规模数据中心时时刻刻都可能发生大大小小的部件失效，因此Dynamo能够提供非常高的可用性。Amazon公司的Simple Storage Service（S3）是一个支持大规模存储多媒体这样的二进制文件的云计算存储服务。Amazon公司的SimpleDB是建立在S3和Amazon EC2之上的用来存储结构化数据的云计算服务。

### 7.1.7 许可证管理与计费

许可证管理与计费是IT基础设施的最终支付环节，涉及到服务提供商与客户的切身利益。客户通过购买许可证或者支付费用获得对软件、硬件、服务的产权或使用权利，以及相应的售后服务支持；各个提供商获得客户支付的费用。因此，通过许可证管理与计费，整个信息技术行业才得以运转。

仅从软件的许可证计费模型来看，在传统的软件许可证购买方式下，用户需要估算自己需要使用的软件的CPU数量、主机数量、用户数量，然后根据软件发售商提供的许可证计算方法，得到一个需要购买的许可证数量的最大值，作为最终购买的数量。举例来说，用户的数据中心有100台机器需要使用一个软件，每台机器有1个CPU，那么用户购买软件时，需要购买100个许可证。但在实际使用时，可能只有几台机器在使用这个软件，而使用软件的机器上的CPU占用率也远远不足100%。也就是说，在传统的软件许可证计费模型下，用户购买了远远超过其真实使用量的许可证数量，可以说是花了不必要的费用。

随着云计算时代的到来，IT基础设施的许可证管理与计费模式将发生重大变化。在云计算的场景下，用户可以按需付费或者按使用计费，少花冤枉钱。在按需付费模式下，用户可以估计自己对于软件许可证的使用情况，决定自己采购的许可证数量。云计算环境会根据用户的支付给用户一定量的许可证，并按照用户在云计算环境中的使用情况计算已使用的许

可证数量或释放许可。当剩余的许可证数量少于某一个特定的阈值时，系统会提醒用户，让用户决定是否追加付费，或者减少他当前使用的许可证数量。在按使用计费的模式下，用户甚至不需提前估计自己需要的许可证数量，系统会自动跟踪用户在云计算环境里的使用情况，定期生成许可证账单。也就是说，未来用户使用云计算环境中的资源，会像使用水和电一样简单方便。虽然云计算的新型计费模型设计得非常美好，但是目前为了达到这个目标还有很多工作要做，其中最迫切的一个问题就是，大量的软件、硬件提供商目前还没有制定出其产品对应在云计算环境下的计费模式，从而成为了这些产品进入云计算环境的障碍。

目前比较成熟的云环境计费模型是Amazon公司提供的Elastic Compute Cloud（EC2）和Simple Storage Service（S3）的按量计费模型，用户按占用的虚拟机单元（固定频率和数量的CPU、固定数量的内存、特定操作系统）、IP地址、带宽和存储空间付费。具体来说，在EC2中，对虚拟机单元的计费分为两类，一类是按需要的虚拟机单元，即用户使用时才生成、部署，EC2不保证该单元一直在系统中存活；另一类是预留的虚拟机单元，该类虚拟机单元一旦被购买，EC2会为该虚拟机预留空间，并根据用户的需求一直保持开机状态。两种计费类型都支持按使用时间计费。

在S3中，存储服务被分为三类：数据存储、数据传输和数据请求操作。S3对数据存储和数据传输按流量计费，且流量越大，单位存储的资费越低。对于数据请求操作，按照请求的类型按次计费：PUT（修改值）、COPY（拷贝值）、POST（增加值）三个占用存储空间的操作，以及LIST（列表）这个比较复杂的操作费用较高，GET（取值）这个最常用的且不占用存储空间的操作，费用为前面几个操作费用的十分之一，而DELETE（删除）这个释放空间的操作不收取费用。Amazon公司通过上面EC2和S3的计费机制已经收到了很好的盈利效果，但是还不能大规模推广到其他的云环境，例如Amazon EC2的计费是以虚拟机为单元的，没有考虑虚拟机内运行的软件及软件的使用情况。

## 7.2 云计算的技术挑战

我们在第1章讨论了传统数据中心面临的一些值得关注的关键问题，例如安全性、可用性、可伸缩性等。在即将进入的云计算时代，这些关键问题又具有了新的内涵，本节将介绍云计算中这些关键问题的特点和研究进展。

### 7.2.1 安全性

在云计算环境中，用户不再拥有基础设施的硬件资源，软件都运行在云中，业务数据也存储在云中，因此云计算安全关系到云计算这种革命性的计算模式是否能够被业界接受。云计算的安全问题主要有两方面：一是云计算自身环境特有的安全问题，二是云计算会怎样改变现有的软件系统安全防护模式。

从第一个方面来说，传统的观念认为将信息保存在自己可控制的环境内，比存放在不了解、不熟悉的地点更安全。因此，云计算在安全领域遇到的第一个问题，就是传统用户无法认可自己不可控的环境能提供更好的安全性。其实，用户的个人电脑或者中小型服务器、数据中心，远没有云计算环境安全。因为在云计算环境中，数据中心和它运行的基础服务都有专业的机构和人员进行运营和管理，他们远比个人用户及中小企业的IT管理员更有安全管理的经验。同时，云计算提供的资源抽象、隔离、用户管理等技术，也能更好地提高安全性。另外，由于云计算提供的规模效应，用户可以在付出更小成本的情况下享受更高级别的安全服务。

不过，云计算还有一些安全问题有待解决。由于云计算最开始是在企业内部网络运行，并不对外开放，因此云计算在设计之初没有太多考虑安全性问题，从而导致云计算安全的一系列问题。首先，传统的IT系统是封闭的，存在于企业内部，对外暴露的只有网页服务器、邮件服务器等少数接口，因此只需要在出口设置访问控制、防火墙等安全措施，就可以解决大部分安全问题。但在云环境下，云暴露在公开的网络中，任何一个节点及它们的网络都可能受到攻击，因此安全模式需要从“拒敌于国门之外”改变为“全民皆兵，处处作战”，而大多数安全厂商还没有准备好迎接这样的场景。其次，在云环境中，用户的服务系统更新和升级大多数是由用户在远程执行的，而不是采用传统的在本地按版本更新的方式，每一次升级都可能带来潜在的安全问题和对原有安全策略的挑战。另外一个严重的问题不是技术层面的，而是政策法规层面的。虽然人们经常把将数据存在云环境中与把钱存在银行中做类比，但是云环境与银行最大的区别就在于，银行业是一个传统的行业，有相应的法规来规范银行的流程和制度，另外国家或者相关机构对银行的信誉进行了担保，而对于云环境来说，目前缺乏有效的规范和立法，云环境提供商的信誉完全依靠于用户的认同感，对云计算环境的规范和立法，也是一个需要关注的问题。

当然，云环境也为安全策略提供了新的思路。例如，传统的病毒防护模式需要杀毒软件在用户本地储存病毒特征库并及时对其进行更新，从而对本地的病毒进行实时监控。用户需要经常从杀毒软件公司的数据库下载最新的病毒特征库，用户之间是相互独立的。而在云计算环境中，用户高度互联，任何一个用户遇到问题，几乎可以实时地发布给云内的其他用户，多个用户可以协同解决这个问题。这样就避免了频繁更新病毒特征库的操作，而且可以直接享受到最新的安全服务。

### 7.2.2 可用性

可用性（Availability）指的是软件系统在给定一段时间内正常工作的时间占总时间的比重，通常用百分比来衡量。在传统的数据中心中，影响服务可用性的因素有服务器异常宕机、服务被攻击、操作系统崩溃、软件崩溃、停电、网络中断等。数据中心管理员需要采用冗余和灾难备份等方式来保证服务的可用性。然而，这些冗余或者灾难备份系统的引入又带来了新的问题，比如冗余备份带来副本一致性问题，以及更高的采购和管理开销。软、硬件设备和系统自身出现问题是不可能完全避免的，云计算高可用性的本质是通过技术创新，保证即使软、硬件出现问题服务仍然可用，比如虚拟化技术提供的快速部署、虚拟机实时迁移能力，都将云计算环境的可用性提到了一个新的高度。

云环境能够在最大程度上减少资源的不可用对业务系统的影响，打造具有高可用性的计算环境。在云计算中，提供对运行时间的保证和服务级别协定已经成为对大多数云计算提供商的标准要求。这些云计算平台大多声称能够提供99.999%的可用性。但实际上，现有的云计算环境也出现过可用性问题，在2008年上半年，Amazon公司的S3云存储服务出现大范围和长时间宕机的情况；Google Gmail等服务也偶尔发生无法访问的现象。这些问题的出现使得人们对现阶段公有云计算产品的高可用性产生了质疑。

在发生物理故障的时候，服务器硬件关机的时间很短，而从备份状态恢复往往需要更长的时间。一个微小的云计算故障可能导致软件故障的连锁反应，从而引起依赖云计算的某个软件服务中断几个小时、十几小时，甚至几天。这就意味着云计算整体环境的可用性也许能够达到99.999%，但是，用户所关注的单个服务或应用的可用性却不能达到99.999%。

为了提供真正高可用的服务，云计算的提供商正在研究常见故障的分

析及预测模型。基于对这些模型的研究，云计算服务商希望能够预测到可能的可用性问题，并通过提前准备复本、提前解决故障、通知用户等手段来避免这些故障的发生，或者减少故障发生带来的损失。

### 7.2.3 可伸缩性

可伸缩性（Scalability）是软件系统的一种特性，具备可伸缩性的软件系统能够通过资源的增加或减少来应对负载的变化，并保持一致的性能。很多传统的应用程序在设计和编码阶段，并没有考虑可伸缩性问题。现代数据中心中的大规模服务在设计之初已经开始考虑可伸缩性问题，并做出了很多有益的尝试。

可伸缩性管理的实现方法主要是垂直伸缩（Scale Up/Down）和水平伸缩（Scale Out/In）。垂直伸缩是指在现有的服务节点上增加或者减少资源，比如增加或减少CPU、内存、线程池和存储空间等。而水平伸缩是指在现有的服务节点基础上增加或者减少服务节点，从而支持更多或者更少的服务请求。水平伸缩需要原有系统提供对多个服务器组成的集群的管理，包括数据同步、统一监控、负载均衡和性能调优等。

在云计算环境中，对于应用的垂直伸缩和水平伸缩都可以通过云计算的基础设施平台得到支持。比如在一个基于服务器虚拟化的云基础设施中，垂直伸缩可以通过对虚拟机的资源调整来实现。虚拟化平台提供了丰富的接口，使管理员可以方便地调整一个虚拟机的CPU、内存或者存储资源。对于水平伸缩，则可以通过增加或减少应用对应的虚拟机节点来完成。在云计算的环境中，应用在理论上可以做到随意伸缩，即应用所占用的资源可以随着负载的上升或降低而增加或减少，从而保证在不同的负载下仍然能获得一致的性能。

云计算对于可伸缩性的要求通常包括及时、适量、细粒度、自动化和预动性。这些要求同样适用于云基础设施。虚拟机的资源调整可以即时生效，保证了可伸缩性管理的及时性要求；资源的伸缩基于应用对于资源的需求，因此保证了可伸缩性管理的适量要求；CPU、内存、存储资源等可以在非常细的粒度上调整，保证了可伸缩性管理的细粒度要求；基于应用性能及资源需求的自动化可伸缩性管理保证了自动化需求；基于应用历史记录、应用模式及预测模型预测出的可伸缩性调整，满足了可伸缩性管理的预动性需求。

## 7.2.4 信息保密

信息保密与信息安全有所不同，信息安全是指信息不会被攻击、篡改，而信息保密是指信息的内容不应该被未经过授权的人得到。

云计算服务商认为，对于云计算、云环境的信息保密问题，用户是可以放心的，因为数据在云的大规模分布式存储机制中，完整的数据实体通常是被打散成一些“块”或者“碎片”存储在不同的服务器上的，每个块甚至包含来自不同数据实体的内容。因此，一个块可能是一个很大的逻辑文件的一部分，也可能包含多个很小的逻辑文件。如果一个非法用户想要窥探云中的数据，他必须获得大量的存储服务器的访问授权，而这个工作是非常困难的。

即便如此，上述方法只是增加了非法用户访问信息的难度，而没有根本解决问题。非法用户可以通过暴力破解所有的存储服务器来收集信息，他甚至会破解云存储系统的数据分发逻辑，从而精准地找到每一个块。同时，多个文件可能共同存储在一个大块里，这增加了数据泄漏的风险。解决这些问题的根本做法是从逻辑上，甚至从物理上将多个用户的数据隔离。

信息保密还需要考虑不同国家相关法律、法规之间的差异。目前，如果同一个云的多台服务器放置在不同国家，它们面对的IT管制政策会有所不同。比如，某些国家要求经过法律授权的机构有权查看存放在数据中心里的数据，而另外一些国家会严格保证用户的数据隐私。由于存在这种区别，用户在使用云服务时，会提出对于数据保密的个性化要求，比如要求数据一定要保存在严格保证隐私的国家的服务器里，或者用户会要求将涉及国家安全、企业利益的敏感信息，必须存放在某个国家、某个网段里，并拥有附加的保密服务。

多家云计算厂商已经关注到了信息保密的问题。IBM公司表示，该公司会制定更多的相关管理流程与技术标准，来保证客户的数据不被泄露。Google公司也正在研究如何处理用户对于数据存放位置的个性化需求。

## 7.2.5 高性能

通过对大量服务器的整合和调度，一个云计算环境能够为用户提供远远超过传统计算环境的计算、存储和通信性能。但是，云环境所承担的

计算、存储和通信方面的负载也远远大于传统的计算环境。不同的云环境所采用的技术可能完全不同，本小节将着重分析当前云环境中最流行的技术的性能，包括服务器虚拟化技术、大规模数据处理技术和分布式存储技术。

服务器虚拟化是云计算基础设施层的重要技术，它的性能会影响整个云中几乎每一个节点的性能，因此虚拟化的性能就成了云计算性能的关键部分。根据多家机构的测试，在目前主流的半虚拟化系统中，例如Xen和VMware ESX，虚拟机管理系统只会带来少量的额外CPU开销。而内存的性能开销问题则较为严重，因为内存作为一个物理机的共享资源，其操作逻辑相对简单，多个虚拟机访问物理机的同一块内存时，很容易出现访问冲突，一旦出现冲突，虚拟机管理系统就要接管虚拟机与物理机内存的I/O操作，并负责内存资源的调度，这个过程叫做陷入。陷入后的内存操作性能远低于虚拟机对物理机内存直接操作的性能。在某些情况下，半虚拟化的虚拟机监视器会产生较大的内存开销。因此，对于现在的虚拟化技术来说，原有的CPU密集型的应用能够比较好地迁移到虚拟化平台，而原有的内存或I/O密集型的应用，例如数据库，就会遇到比较大的性能问题。

作为云计算大规模数据处理的事实标准框架，MapReduce也存在着性能问题。首先是适用性导致的性能问题，由于Google公司设计的MapReduce主要针对的是Google搜索引擎的索引、搜索、排序等服务，并不是从完全通用的出发点考虑的，因此MapReduce在使用中存在着适用性的问题。其次，MapReduce的原语设计也会导致性能问题。伯克利大学最近发表的一篇针对MapReduce的论文认为，如果将MapReduce作为一种通用的数据处理框架，它相对于MySQL提供的数据处理操作，还缺少一个叫Merge的原语，因此论文提出了一种叫做Map-Reduce-Merge的改良计算模型，意图通过提供Merge原语，来提高MapReduce的效率。从MapReduce算法的流程来看，Reduce操作需要等大部分Map操作完成才能够继续，如果Map操作耗费非常长的时间，那么Reduce操作会一直等待。在某些情况下，采用MapReduce比集中的数据处理并不会快多少，出现这种情况的原因可能是计算的分布不均匀，或者某些Map节点的计算能力远远低于其他Map节点。此外，由于MapReduce运行在分布式系统上，系统中的节点通过网络进行连接，因此在MapReduce运行过程中需要大量的网络消息通信，比如一个MapReduce计算环境中有M个Map节点和N个Reduce节点，那么可能的通信链路就有M×N条，这些链路上的数据交换会给网络带来

大量的负载。目前大部分数据中心的网络架构还是基于共享带宽、中心交换和路由的，而不是理想的点对点连接方式，因此较高的负载会带来额外的通信开销，甚至影响到其他节点之间通信的性能和可用性。以上这些问题，都是采用MapReduce框架，或者实现MapReduce技术的用户需要考虑的技术挑战。

另外，分布式系统常用的分布式存储在云计算环境中面临着更严重的性能问题。相对于原有的本地存储、集中存储或者网络存储，云计算分布式大规模存储面对的是一个网络不可控的环境。对于典型的MapReduce加上分布式存储的云计算场景，每一个Map或者Reduce计算节点都需要从分布式存储中读取或者保存数据，由于计算网络与存储网络是分别搭建的，绝大多数情况下计算所需要的数据存取操作都不会发生在计算节点的本地，而是被分发到分布式存储网络的其他节点上。在最坏的情况下，一个计算节点及其对应的存储节点可能分别处在地球的两端，这会导致严重的性能问题。解决这个问题的方式之一是，对数据安全性、可用性和数据同步要求不高且数据量比较小的应用采用本地文件系统，对规模比较大的应用采用可以感知位置甚至上层应用的网络存储系统，这样就可以减少上述性能问题的发生。





### 7.2.6 标准化

如果用户希望维护多个云之间的数据同步、应用版本同步，或者应用在多个云之间的互操作，那么最理想的情况是通过一种方法将多个云数据中心抽象成一个，以此来降低使用的复杂性。根据业界多年来的经验，这个工作只能通过标准化来完成。

云计算技术目前还在起步阶段，关于云计算的标准化工作还在酝酿之中。按照常规的技术生命周期，一般一个技术要在出现一个或几个市场占有率较高的厂商之后，才会在这些厂商的带领下制定出相关的技术规范和标准。目前云计算还在发展初期，还没有开始进行正式的标准化工作，但是众多厂商已经在朝着这个方向努力了。

在云的基础设施领域，虚拟化的主要厂商之一VMware在发起一个叫做vCloud的接口规范，这个规范希望通过对基于虚拟化的数据中心（企业里的私有云）与其他“云”（其他企业的私有云、公共云）的接口进行标准化定义，来抢占云计算标准化的先机。在2008年9月举办的VMworld大会上，

VMware公司宣布了它的vCloud计划，介绍了多家支持这个计划的厂商，并希望虚拟化的平台使用者、开发者甚至整个业界都来加入这个计划。VMware公司引领vCloud标准的优势在于它是虚拟化领域的先进厂商，拥有大量的合作伙伴，以及一个包括了大量应用的虚拟器件仓库。从本书的讨论中我们已经了解，虚拟化虽然是云计算重要的组成部分，但并不是云计算的全部，VMware公司在制定标准时如果过多地从虚拟化角度及自身产品的角度来考虑问题，可能会导致应用云及平台云厂商在vCloud计划中找不到自己的位置。

云计算标准化领域的最新进展是在2009年3月，以IBM、思科、SAP、EMC、RedHat、AMD、AT&T、VMware为首的近百家IT公司联合发布了“开放式云宣言”（Open Cloud Manifesto）。这个宣言总结了云计算的特点和现有的挑战，并明确提出了建立开放的云基础设施将是未来云计算领域的发展趋势。云计算服务应该成为公共事业，并作为服务提供给用户。用户可以方便地在多个云之间迁移数据和应用，可以快速、敏捷地开发新的应用，并减少学习云计算技术的难度和时间。涉足云计算领域的厂商应该以完全开放的心态来开发自己的技术，而不是通过隔离的技术来割裂用户群、增加云计算推广的难度。这个宣言的目的并不是推出一个具体的标准，更多的是对一个开放标准的呼吁。同时，这一宣言也考虑到了常见的标准化过程中出现的问题，即多家厂商都推出各自的标准，导致最后没有任何一个标准能够被广泛接受。因此，在云计算领域，该宣言希望所有厂商都能在这个宣言提出的纲领下进行对话和协商，推出统一的、简单的充分参考并建构于已有技术标准之上的云计算标准。希望那个在不远的将来将会诞生由上百家业界厂商共同制定的开放的、统一的云计算标准。





## 7.3 小结

本章概括叙述了云计算产生、发展、推广过程中的新技术，它们包括快速部署、资源调度、多租户技术、海量数据处理、大规模消息通信、大规模分布式存储、许可证管理与计费模型等。同时，我们分析了云计算给传统数据中心的安全性、可用性、可伸缩性、信息保密、性能和标准化等问题带来的新挑战。在每一个关键技术的论述中包括了其产生的背景、要解决的问题、目前发展的现状、主流的解决方案，以及需要研究的关键技术。本章有助于读者对云计算技术有更深的了解，能够亲身去体验云计算的这些关键技术。另外，在看到云计算美好前景的同时，读者也可以了解云计算目前的不足和面临的挑战。

CHAPTER 8

# 第8章 云计算的业界动态

自2007年开始，业界对云计算的研发、讨论，以及相关产品的关注持续升温。云计算为人们展示了未来信息技术的发展趋势，也为业界带来巨大的潜在商业价值。根据市场研究公司IDC的预测，到2012年，云计算市场份额将占据整个IT市场份额的1/4。本章将介绍业界从事云计算研发的重要公司及它们的产品，帮助读者加深对它们的了解。

## 8.1 IBM

IBM是一家业务涵盖硬件、软件、咨询和服务的综合信息服务公司，是云计算的重要倡导者和推动者。IBM公司的云计算战略全面融入了IBM随需应变（On Demand）的思想、面向服务架构（SOA）的设计和动态基础设施（Dynamic Infrastructure）的理念。目前，IBM公司已经提出了完整的云计算战略，组织参与制定云计算的相关规范，并积极探索云计算的新型商业模式。IBM公司认为云计算能够提供一种经济有效的业务模型来部署应用和管理服务，可以帮助客户改善服务质量、降低成本和控制风险。该模型简化了服务的交付和管理流程，提高了信息化基础设施对业务需求的响应速度。凭借在虚拟化、标准化和自动化方面积累的经验和雄厚的技术实力，IBM公司正在为不同的客户量身打造适合他们的云计算解决方案。从提出云计算的概念至今，IBM公司已经帮助分布在全球十多个国家和地区的数十家企业和机构搭建了自己的云计算环境。IBM公司在全世界范围内推广普及云计算技术，帮助更多的人了解云计算。此外，IBM公司十分注重与学术界的合作，目前正在和全球多所顶尖高校进行云计算的合作研究。

IBM公司在云架构的每一层都提供了整合的解决方案来帮助客户设计、构建和管理云环境，保证客户业务在云环境中的高效运转。本节将分

别介绍IBM公司在云计算基础设施层、平台层和应用层的主要产品和解决方案。

### 8.1.1 概述

在基础设施层，IBM公司采用虚拟化技术将计算、存储与网络等硬件封装起来，形成一个统一的资源池，以服务的方式为上层用户提供资源，这些用户可以是服务使用者或者是应用服务。IBM公司基础设施层的硬件产品有：具备服务器虚拟化能力的z系列服务器、p系列服务器和x系列服务器，以及具备存储虚拟化能力的SAN Volume Controller等。前者通过服务器虚拟化将计算资源抽象为虚拟服务器，方便上层对这些虚拟服务器的整合和管理；后者将来自不同物理设备的存储空间抽象为单一的资源池，统一管理存储设备，提高存储资源的利用率。

基础设施层硬件资源的虚拟化是实现资源整合和优化的基础，如何更好地管理这些虚拟化资源是体现云计算优势的关键所在。为此，IBM公司提出了Ensembles的概念，用于消除物理资源之间的边界。Ensembles的底层是一组由网络连接的物理资源，它们通过虚拟化技术被抽象为资源池。不同于被虚拟化前的相互分离的物理设备，这个资源池是一个可扩展、可管理的单一系统，通过统一的接口向上层提供服务。Ensembles具有自下而上贯穿整个基础设施层的管理能力，负责物理平台、虚拟化层和资源池中资源的规划、创建、组装和调整等任务。

如何合理规划应用所需的资源、搭建应用的云基础设施环境、将应用部署到运行平台使其成为可用的解决方案、对解决方案进行持续的生命周期管理，这些都是基础设施层需要考虑的问题。为了应对这些需求，IBM公司推出了Tivoli Service Automation Manager（TSAM）。该产品内置了丰富的解决方案模板，帮助用户构建自己的云基础设施环境和应用运行环境。设计阶段完成后，TSAM通过工作流的方式简化复杂的部署逻辑，使应用快速可用。在运行阶段，TSAM自动检查预定义的管理计划，对应用进行日常维护并检查维护效果。当应用完成其任务，生命周期结束时，TSAM会自动回收该应用使用的资源，将它们返还给底层。

平台层利用基础设施层提供的资源，为用户提供应用开发、部署、运行和管理等服务。为了提高云应用的开发效率，IBM将Rational家族的开发工具产品改造升级，使其能够适用于云计算的场景。例如，Rational

Application Developer（RAD）可以帮助用户快速地规划、分析、设计、开发、测试基于Java等编程语言的Web服务和门户应用程序。

WebSphere是IBM公司五大软件品牌之一，提供了构建、运行和整合SOA应用的中间件平台。WebSphere CloudBurst Appliance（WCA）是WebSphere的云计算平台层产品。WCA通过虚拟器件技术显著降低了构建平台云的复杂度，大大缩短了构建时间，为云应用提供了一个自动、高效、可靠、可伸缩的SOA中间件运行环境。WCA采用模板机制，将WebSphere多年来的工程经验和客户反馈融汇到虚拟器件的创建、组装、部署、激活、监控和管理维护中去，保证了所构建的SOA环境的优化运行。

在应用层，IBM公司提供了丰富的面向企业用户的云应用，如帮助用户进行在线协作的LotusLive应用。LotusLive为企业用户整合了在线协作服务和社会网络服务，用户可以在LotusLive平台上随时随地与同事和客户进行在线会议、文件共享、即时通信和项目管理等任务。无论用户是在远程工作，还是在管理远程的团队，LotusLive都可以通过网络将同事们聚集到一起，在安全的应用环境中提供全套的协作解决方案。

云计算作为一个逐步普及的新型计算模式，需要不断地引入最新技术研究成果。在IBM的众多云解决方案中，我们选择性地介绍了已经成功应用于IBM全球八个研究院的IBM RC2解决方案和与我国的具体实践相结合的IBM云环境管理解决方案。针对云环境中解决方案的规划、构建、部署等生命周期管理和各个阶段所面临的主要挑战，这些解决方案在充分了解客户需求的基础上结合了IBM在虚拟化和云计算领域的重要研究成果并提供了完备的技术解决方案。

除了在云计算领域提供相关的软、硬件产品和解决方案，IBM公司还提供咨询服务，帮助客户充分理解和有效利用云计算所带来的优势。IBM公司为来自不同行业的客户提供了针对其行业特点的云计算咨询服务，协助客户评估总拥有成本，设计适合特定需求的云计算场景，搭建自己的云计算环境，完成从传统环境到云计算环境的迁移。咨询服务与IT软、硬件产品和服务相辅相成，是IBM公司完整云计算战略的重要组成部分。

### 8.1.2 IBM Ensembles

简化IT资源管理是云计算基础设施层的重要目标。只有将数量众多、

分散的物理系统整合成为一个大规模、一致的资源池，才有可能实现简化管理。但是，由于基础设施硬件的多样性和分散性，怎样屏蔽底层的物理差异、为上层提供资源抽象就显得尤为重要。

作为IBM公司云计算战略中重要的基础设施层方案，Ensembles是一组采用虚拟化技术实现的资源池，主要包括计算资源池—服务器Ensemble、网络资源池—网络Ensemble和存储资源池—存储Ensemble。虚拟化技术隐藏了底层的技术细节，提供了对资源的管理、配置和调整功能。在这三种类型的Ensemble之上的是Ensemble服务接口，它为用户提供统一的操作接口。Ensembles通过虚拟化技术整合了数据中心的服务器、网络和存储资源，如图8.1所示。每种类型的Ensemble都具备独立的管理功能，针对各自的资源类型完成资源的加入、释放和维护等操作。Ensemble服务接口为上层用户提供了访问、获取、返回不同类型Ensemble资源的能力。上层用户可以是最终使用者，也可以是应用服务。

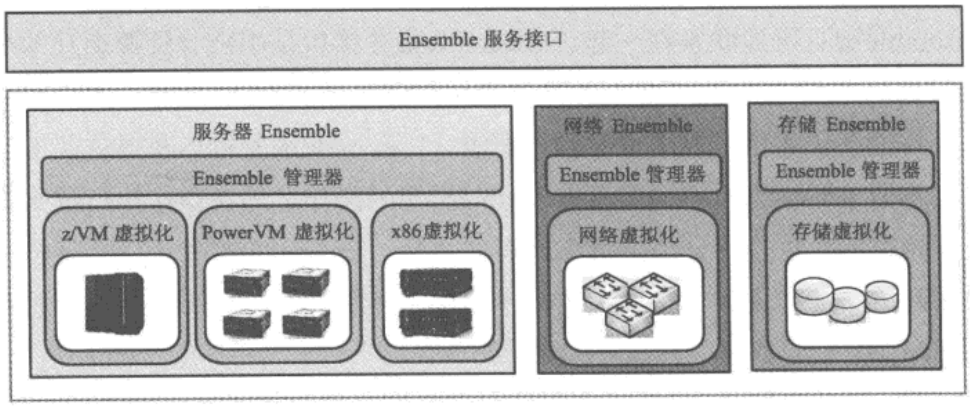



图8.1　IBM Ensembles

服务器Ensemble是一组由同构物理服务器组成的计算资源池，它还包含了管理物理服务器和虚拟服务器的功能。服务器Ensemble中所有的物理服务器都采用虚拟化技术被整合成为一个虚拟服务器集合。服务器Ensemble要求物理服务器彼此兼容，通常采用单一的处理器架构，如x86、Power或者z/Architecture。

存储Ensemble将一系列分散的存储设备抽象成一个统一的存储资源池，它可以访问并整合多种类型的存储服务，包括物理设备（如Logic Unit Number，LUN）、共享存储（如Network Attached Storage，NAS和General Parallel File System/Scale-Out File System，GPFS/SOFS）和虚拟存储（如SAN Volume Controller，SVC）。

网络Ensemble是由一组网络资源构成的统一实体，包括交换机、路由器、VPN网关等网络设备。网络Ensemble实现虚拟的网络连接功能，为用户提供创建网络连接、IP路由与过滤、负载均衡与监控等服务。

无论是服务器Ensemble、网络Ensemble还是存储Ensemble，都必须实现关键的管理操作，如用户账户、资源操作、效用优化、安全管理等。这三种类型的Ensemble中都内置了一套灵活的软件管理框架。如图8.1所示，Ensemble管理器将这些管理操作整合到一起，封装为统一的接口提供给Ensembles以外的实体。Ensemble管理器是Ensembles的关键组件，负责每个Ensemble的系统管理，如工作负载优化、可用性保证、系统启动和关闭、软件恢复和更新等操作，同时它还负责硬件资源的管理，如热量控制和能耗监控等。

Ensemble服务接口将以上三种Ensemble整合为能够被统一访问和管理的基础设施资源池。其中，服务器Ensemble和存储Ensemble通过网络Ensemble被有机地联系在一起。Ensemble服务接口是用户与资源交互的平台，用户通过它获取可用的资源列表，制定资源的使用策略、参数及选项，并根据这些信息发出资源操作请求。Ensemble服务接口获得这些请求后，将请求转换为对不同类型Ensemble资源的操作，并在操作执行过程中进行持续的监控，保证用户请求的有效执行。Ensemble服务接口还具备一系列高级功能，通过分析应用需求来提高服务质量。应用需求是指应用对Ensemble类型的依赖、对资源质量的要求等。

总之，在IBM云体系结构中，服务器、存储和网络等资源被封装为Ensembles。上层应用和解决方案的正常运行，需要服务器Ensemble提供的计算资源，需要存储Ensemble提供的存储资源，需要占用网络Ensemble提供的带宽使得这些资源互联互通。Ensembles作为IBM云架构中的基础设施层，为上层提供资源访问和管理的接口。这样，平台层就可以通过多种工具，如Tivoli Provisioning Manager（TPM）和IBM Tivoli Monitor（ITM）等来访问基础设施层的服务。Ensembles隐藏了资源内部的实现细节，使用者只需关注所需资源的类型和数量，而不需要直接对它们进行操作。可见，Ensembles简化了IT基础设施资源的获取和使用方式，减少了云计算不同层次之间的耦合程度，使整个云架构具有了更好的可扩展性和灵活性。

## 8.1.3 IBM TSAM

简化应用运行平台的管理是云计算基础设施层的重要目标。通常情况下，一个完备的管理平台除了要对基础设施层资源进行操作和管理外，还要处理平台内部各种软件之间的关联关系。这样的关联关系往往烦琐而复杂，需要平台层使用者或管理员具有丰富的知识和经验。这种对平台层组件的管理贯穿上层应用的整个生命周期，管理操作是持续而且不可间断的。这些因素制约着云计算基础设施层充分发挥其效用。

IBM Tivoli Service Automation Manager（TSAM）为用户提供了管理应用服务生命周期的方案。TSAM帮助不同角色的用户按照ITIL V3的最佳实践经验来管理服务的生命周期。作为云计算管理服务的重要产品，TSAM担当着云管理者和协调者的重要角色，既要对云架构中各种产品进行完整生命周期的管理，又要通过调配和优化资源满足客户对服务质量的要求。TSAM的设计强调更快速的服务响应和交付能力，以及更低的运营成本。TSAM提供了三个阶段的管理功能，包括服务的设计阶段、部署阶段和运行时管理阶段，支持两种角色的用户，它们是服务设计者、服务运营和管理者，如图8.2所示。

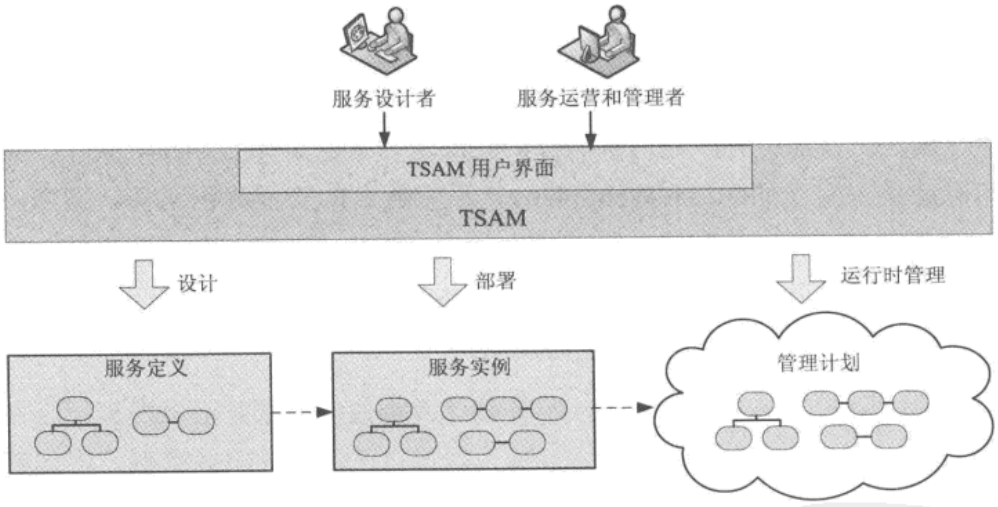



图8.2　IBM TSAM

在设计阶段，服务设计者通过服务定义（Service Definition）来设计服务。TSAM提供了丰富的预置服务定义来简化服务设计者的工作。服务定义规范了对特定环境进行管理的总体框架。TSAM提供三种服务定义：操作系统器件服务（OS Appliance Service）、自助虚拟服务器部署（Self-Service Virtual Server Provisioning）和解决方案服务。

操作系统器件服务主要针对IBM的大型主机（Mainframe），该服务定义描述了在该系统平台上进行服务管理的全套流程，涉及由镜像创建z/VM虚拟服务器、运行虚拟服务器上的软件器件（Software Appliance）及对虚拟服务器和软件器件的运行时管理操作。这些z/VM虚拟服务器被创建在一个或多个安装了z/VM逻辑分区（LPAR）的宿主平台上。在TSAM中，软件器件包括操作系统，以及其上可选的软件，如DB2、性能监控软件等。

自助虚拟服务部署主要针对IBM的System x和System p服务器，在由这些服务器构成的数据中心里提供对虚拟服务器和相关软件的全套流程管理。通过一组简单且易于操作的工具集，用户可以选择他们所希望得到的软件栈，将软件栈安装到虚拟服务器中，并实现虚拟服务器在物理环境中的自动部署。在TSAM看来，每个解决方案服务都是由多个虚拟服务器上运行的应用连接而成的。为了实现对服务的管理，自助虚拟服务部署定义了以下操作：（1）通过创建虚拟服务器及其上的软件栈来创建新的服务，或者为现有服务加入新的虚拟服务器；（2）为虚拟服务器安装软件栈，包括操作系统和中间件；（3）销毁一个虚拟服务器并释放其占用的资源；（4）为虚拟服务器增加或减少资源，比如CPU和内存；（5）销毁一个解决方案服务并释放占用的资源。通过这些操作，用户可以便捷地在云计算环境中创建、管理和销毁服务。

解决方案服务在以上两种服务的基础上，提供了针对不同中间件、应用和解决方案（如DB2和WebSphere产品家族）的管理流程定义。例如，WebSphere集群服务（WebSphere Cluster Service）是TSAM的可选组件，它定义了如何在云计算环境中部署IBM WebSphere的应用服务器产品，将它们配置成集群并管理该集群的生命周期。

完成服务定义的设计之后，服务设计者将他们的服务定义发布到服务定义目录中。服务管理者查阅该目录，选择自己需要的服务类型，将部署请求提交给TSAM。TSAM根据服务定义中描述的部署流程，解析服务各个组件间的依赖关系，根据当前实际情况规划工作流程，为用户暴露出必需的配置选项，获取所需的资源，完成自动部署操作。此时，服务定义已经被实例化为正在运行的服务部署实例（Service Deployment Instance）。

在运行时管理阶段，对服务的日常管理操作主要由服务管理者负责。TSAM为服务管理者提供了管理计划（Management Plan）来实现管理操作

的自动化。TSAM能够自动分析并执行管理计划中每个操作的具体步骤，确认操作的结果，规划下一次操作的内容。比如，当服务管理者决定为现有服务加入一个WebSphere应用服务器时，TSAM决定这个新的应用服务器将怎样被初始化、放置在哪里、是否与其他管理软件关联等。当每一项细节都被敲定，并确认系统当前可以实现这些操作后，这个WebSphere应用服务器才能被TSAM加入到现有服务中。

最后，当服务的生命周期结束时，TSAM回收该服务所占用的资源，并把它们释放。虽然服务已经停止，但是有关这个服务部署实例的信息，如软件栈配置、组件关联关系及运行的历史日志等，仍然可以被TSAM存档，以便日后的查询和审计。

总之，TSAM提供了在多种平台上规划、创建和管理IT资源，并将这些资源整合为可用的云计算能力。通过可选的WebSphere组件支持，TSAM可以在云计算环境下部署、管理SOA应用服务环境。通过与基础设施层的配合，TSAM实现了从硬件到操作系统再到中间件的整体自动化管理。

### 8.1.4 IBM WebSphere CloudBurst

IBM WebSphere CloudBurst Appliance（WCA）是IBM公司中间件软件品牌WebSphere旗下的一款用于创建、部署和管理私有WebSphere云环境的产品，它能够帮助用户创建和管理面向服务的私有云平台，其最大优势在于有效整合了云基础设施层和云平台层。

如图8.3所示，WCA在物理上是一个具有运算、存储和联网能力的硬件器件（Hardware Appliance），该硬件器件包含了WCA软件功能模块和WebSphere虚拟器件镜像与模板。WCA软件功能模块主要具有基础设施管理、解决方案部署、用户群组管理、镜像模板管理、脚本包管理及监控与计费等功能。WebSphere虚拟器件与模板是WCA采用虚拟器件技术快速部署WebSphere环境的基础，WCA利用模板机制将领域专家的经验融汇到WebSphere环境的虚拟化部署过程中，这些模板体现了WebSphere配置的最佳实践经验。从在单个虚拟服务器上运行的简单的单节点环境，到在多个虚拟服务器上运行的复杂的WebSphere集群，用户都可以重用模板，简化WebSphere环境的规划与部署工作。

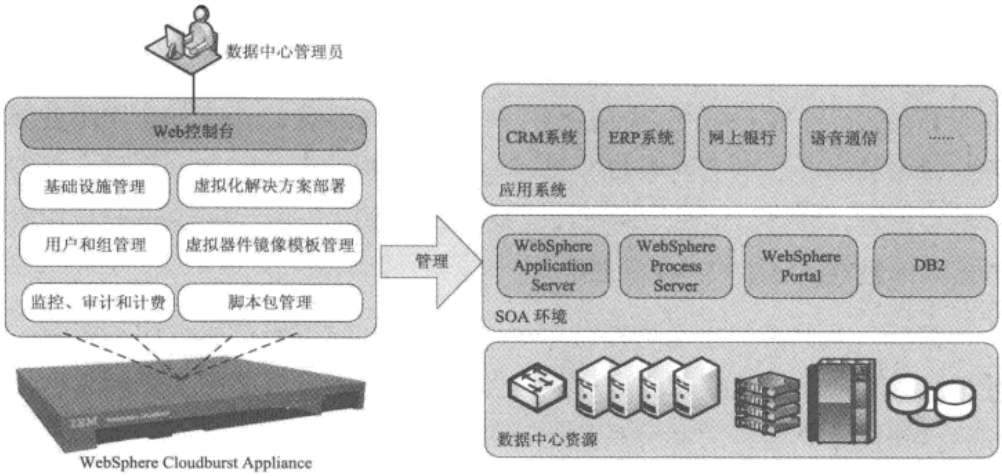



图8.3　IBM WCA

WCA的硬件器件上没有复杂的操作面板，只有一些简单的通信接口，如网口和串口。在绝大多数情况下，WCA通过网口与外界通信，只有在维护模式下才使用串口。这种“即插即用”的方式方便了用户构建自己的私有云。用户首先将WCA硬件接入私有数据中心，再将安装了虚拟化平台的物理机注册到WCA，WCA就可以统一管理这些虚拟化平台，在它们之上部署WebSphere环境。在部署过程中，用户首先选择需要的模板，进行必要的配置，然后WCA自动把虚拟器件镜像发送到目标虚拟化平台，创建虚拟机，激活WebSphere环境。在WebSphere环境的运行过程中，用户可以通过WCA的管理功能对该环境进行持续的监控和优化。

用户可以通过三种方式访问WCA提供的管理功能：Web方式、命令行方式和REST方式。WCA具有一个用户友好的Web 2.0风格控制台界面，它集中了WCA所有的管理功能，用户只需要在页面上进行简单的操作就可以实现复杂的WebSphere环境管理操作。在控制台界面首页，用户还可以下载WCA的命令行工具，通过该工具直接连接到WCA上进行操作。REST方式是为了方便将WCA的软件功能与其他产品进行整合而设计的，其他产品只需要遵循REST编程接口规范，就可以调用WCA提供的所有功能。

用户可以把其私有数据中心的基础设施资源注册到WCA，这些资源包括网络、存储和安装了虚拟化平台的服务器，通过在WCA界面上输入虚拟化平台的名称、主机名、用户名和密码就可以将其纳入WCA所管理的环境中。WCA支持的虚拟化平台有IBM z/VM、IBM PowerVM服务器、VMware ESX和ESXi。用户可以通过WCA的管理界面查看虚拟化平台的实时性能参数和其他相关信息，比如CPU和内存的使用情况、虚拟机的运行情况、关

联的网络和存储的情况等。WCA管理的网络资源包括IP地址、子网地址、子网掩码、网关和DNS。WCA所管理的存储既可以是虚拟服务器的本地存储，也可以是通过SAN和NAS组成的存储网络。

WCA内部的WebSphere虚拟器件是WebSphere推出的另一个产品：WebSphere Application Server Hypervisor Edition。该产品的软件栈包含了基本的操作系统、WebSphere应用服务器、IBM HTTP服务器和激活引擎（Activation Engine，AE）。虚拟器件按照OVF格式打包，保证了兼容性和标准化。

用户可以通过WCA管理虚拟器件镜像，修改某个用户对该镜像的访问权限或导入新的虚拟器件镜像。在虚拟器件镜像的基础上，WCA还预置了一系列可重用的WebSphere模板。这些模板是基于WebSphere产品十余年的最佳实践和用户反馈而创建的，目的是为了方便用户部署相同或者类似的虚拟化WebSphere解决方案。模板的拓扑结构包括了从简单的WebSphere孤立应用到复杂的WebSphere应用集群。用户可以通过WCA的控制台界面获得关于这些模板的创建者、创建日期、被部署的实例及它们的配置等详细信息。另外，用户也可以基于现有的虚拟器件镜像或者模板快速定制自己的模板，这些定制的模板被保存到WCA中，和预置模板一起被统一管理。

脚本包是WCA的重要组成部分，它由一组脚本程序组成，支持用户个性化定制被部署的虚拟化解决方案，如在中间件平台上部署客户应用。用户可以将离线创建的脚本包上传到WCA中，然后配置脚本包的环境变量、工作目录、访问权限及模板绑定等参数。

部署虚拟化解决方案是模板实例化的过程。首先，用户在模板列表中选择所需的模板，然后提供WCA要求的一些非常简单的配置参数，如虚拟机的CPU数量、内存大小、登录密码等，就完成了对虚拟化解决方案的定制工作。定制完成以后，WCA将这个模板实例化，将其部署到用户的私有云中，成为运行的虚拟化解决方案。这个过程有点类似我们所熟悉的面向对象编程模式，WCA中提供的模板就如同我们编写的一个类，用户对模板的配置过程好比通过构造函数传参数给这个类，WCA将这个模板部署到私有云的过程就好比类被实例化为对象，分配了内存空间。

WCA支持多种用户和组角色，并为不同的角色提供不同的权限与操作界面。比如，普通用户和组只有部署他能访问的模板的权限，管理员用户既有普通用户的权限，又有创建和修改模板的权限，还可以管理私有云环境，向WCA添加和删除虚拟化平台，并从控制台界面可视化地获取这些虚

拟化平台的信息。

WCA提供了全面的监控、审计和计费功能。在虚拟化解决方案部署完成以后，用户可以通过WCA界面实时查看虚拟化解决方案的状态和资源利用情况，查询相关的日志信息，获取虚拟机状态和其在云中的位置。除此之外，用户可以通过WCA控制台界面提供的链接远程访问虚拟机和WebSphere环境本身的管理控制台界面。在后台，WCA记录每个用户的操作，并以审计日志的形式保存下来，管理员可以通过WCA提供的界面随时下载这些原始的审计日志。另外，WCA还提供了日志分析和可视化的功能，管理员通过该功能可以查看资源的整体使用情况和每个用户使用资源的情况。在计费管理方面，WCA提供了ILMT（IBM License Metric Tool）脚本包，里面包含了IBM专门用于软件许可证统计的工具ILMT代理（ILMT Agent）。该脚本包负责在虚拟化解决方案部署以后，将ILMT代理安装在虚拟机上，记录它们在运行过程中使用软件许可证的情况，并定期汇报给ILMT服务器。WCA还提供管理自身硬件器件的功能，比如停止和重启器件、检查温度和磁盘容量等。

总之，WCA可以被便捷和快速地部署到用户现有的数据中心，为其提供安全可靠的私有云计算环境，并显著节约企业用户的投资和运营成本，实现更高的投资回报率。



### 8.1.5 IBM LotusLive



IBM LotusLive是IBM公司云计算应用层中软件即服务的典型代表，它是一组通过Web方式交付的服务，包括会议服务、办公协作服务和电子邮件服务三个部分。

会议服务包括两部分：LotusLive Meetings和LotusLive Events。LotusLive Meetings是一个整合了语音和视频功能的在线会议服务。LotusLive Meetings具有很多的优势：从易用性来讲，它向用户提供了简单、便捷的Web操作界面，允许用户自由创建或参加会议；从管理便捷性来讲，它具备了对整个会议的单点管控功能，保证了视频和语音的服务质量；从安全性来讲，它采用了HTTPS安全链接，采用128位密钥加密机制，保证了会议的安全性和私密性；从用户身份管理来讲，每个用户具有单一永久的用户身份，允许用户通过不同的客户端以始终相同的身份创建和参加会议。LotusLive Events是一个在线事件管理和网络会议服务，它不仅包括了LotusLive Meetings的全部功

能，还包括一系列的增强功能，如自动化邮件公告、注册管理、事件预演、事件存档、多浏览器和平台支持等，可以方便地组织联机事件。LotusLive Events为在线会议和事件管理提供了全面的支持。

办公协作服务也包括两部分：LotusLive Engage和LotusLive Connections。LotusLive Engage是一个整合的社交网络模式的协作服务，具备联系人管理、在线会议、档案分享、即时通信、精简版的项目管理等功能。它通过开放的标准整合各项Web服务，同时与既有的桌面应用兼容，如Lotus Notes等。在企业或组织中，用户通过订阅的方式获得该服务，与其他相关人员协作。例如，一个Web网站开发团队中的架构人员、开发人员和测试人员皆可通过该服务进行在线会议，讨论设计的变更、分享设计档案、密切追踪项目进度。与LotusLive Engage相似，LotusLive Connections也提供了集成的社交网络协作服务，并更加强调社交网络对协作效率的贡献、面向文档的资料共享和活动管理等功能。LotusLive Connections帮助企业用户在低风险和低成本的情况下快速集成新兴技术。

电子邮件服务包括基于客户端的LotusLive Notes、基于Web的LotusLive iNotes和一系列附属插件，如LotusLive Mobile和LotusLive Sametime Instant Messaging。LotusLive Notes是一个富客户端电子邮件系统，能够支持较大规模的企业和机构。LotusLive Notes帮助使用者关注高优先级工作、有效地共享信息、迅速地做出决策。LotusLive Notes支持全文搜索、邮件过滤和排序、会话视图与标签等功能，可以对不断增多的邮件进行有效管理。LotusLive iNotes是一个基于Web的安全的电子邮件服务，向用户提供邮件收发和日程管理功能。LotusLive iNotes主要有以下五个特点：（1）全面支持目前流行的邮件协议，比如POP3、IMAP4和具有认证机制的SMTP；（2）日程管理支持可配置的事件提醒和工作日视图功能；（3）有效防止垃圾邮件，提供防病毒功能；（4）安全的SSL加密网络传输；（5）用户账号管理和全局设置。LotusLive Mobile是一款针对手机平台的Lotus协作功能产品。手机用户可以通过该插件访问LotusLive提供的电子邮件服务。LotusLive Sametime Instant Messaging可以在LotusLive提供的电子邮件服务基础上集成即时通信功能，从而方便企业用户进行即时信息共享与办公协作。

### 8.1.6 IBM RC2

为了满足大规模的计算任务对资源的需求，IBM公司分布在全球的

八大研究机构共同创建了一个基于Web的私有云：IBM Research Compute Cloud（RC2），如图8.4所示。经过几年的发展，RC2现在能够为IBM公司全球3000多名研究开发人员以服务的方式提供随需应变、自助获取的计算资源，成为了云计算创新的试验床。

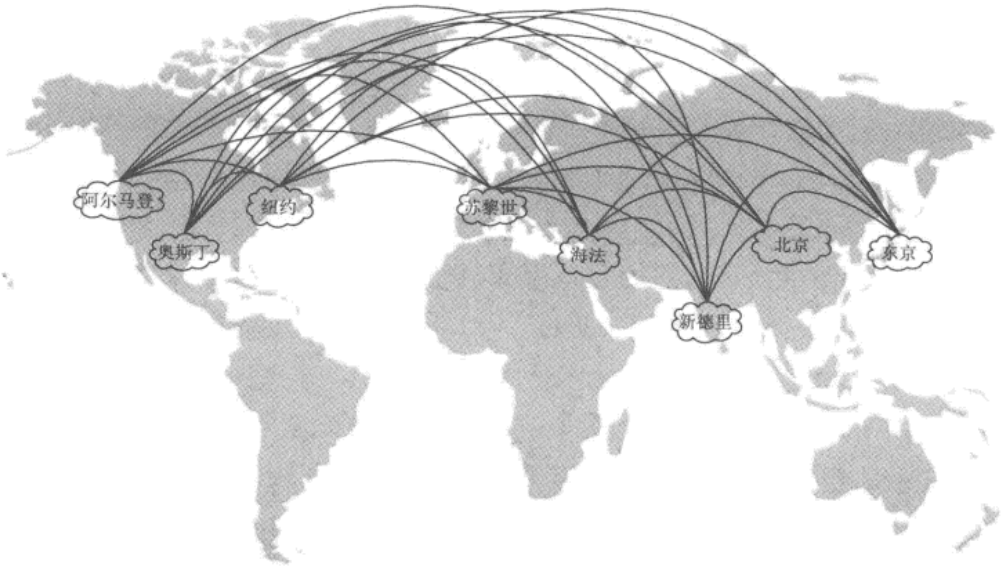



图8.4　IBM RC2

RC2提供了解决方案随需应变的自助交付，支持对服务整个生命周期的管理，包括服务的创建和订阅、资源的动态整合、服务监控和计费等，它基于业务流程机制灵活处理用户请求。RC2支持虚拟化服务环境，可以基于SOA架构整合现有IT资源和产品，具有良好的灵活性。

RC2的诞生源于IBM公司对IT资源的以下需求。（1）更低的管理成本：目前购买设备和软件的费用趋于稳定，但管理成本和能源消耗却在不断增加，这样的情况同样出现在各个部门。（2）应用的快速部署和自助服务：传统的应用部署是一项费时费力的工程，不仅要考虑软、硬件之间的兼容性，还要进行各种繁杂的安装和配置，而应用需要快速多次地部署到相应环境中进行验证，否则将会影响应用的顺利交付。（3）高可用性、高利用率、安全和节能：这是人们对IT资源的普遍要求。（4）以服务的方式管理基础设施：IBM研究部门的实际情况是IT资源分布在全球各个实验室，如果将基础设施的管理功能通过服务的形式交付给管理员，那么将大大简化管理复杂度，而且管理员能够在任何地点通过网络进行有效的远程管理。

作为一个支持复杂研究业务的云计算平台，RC2支持的研究方向包括

虚拟化环境、云存储系统、互联网规模的数据中心和探索性云计算研究。虚拟化环境的主要研究对象是虚拟化的硬件资源及这些资源的管理（虚拟镜像管理、虚拟资源移动性管理、虚拟资源优化整合管理等）。云存储系统是针对大规模存储系统的架构和文件系统的研究，从而得到云计算中存储的最优实现。互联网规模的数据中心着重研究未来分布式数据中心的架构及对供电和空调设备的优化配置。探索性云计算研究关注的是用于服务交付的下一代基础架构，它旨在提供革命性的基础架构，在这个基础架构中，资源和服务以透明和动态的方式被管理、部署和重新分配。

对应于云计算的三层架构，RC2自底向上的具体实现是虚拟化基础架构、业务服务平台和业务流程管理。虚拟化基础架构中的IT资源被虚拟化技术整合成资源池，按照状态的不同，分为可用的资源池、预留的资源池和使用中的资源池。在业务服务平台层，RC2关注资源容量管理、部署、调度、监控、资源使用计量和计费等“云中间件”功能。业务流程管理层是通过服务门户直接面向用户的，通过这一层提供的功能，用户可以使用RC2提供的IT资源和服务。RC2是IBM公司的一个发展中的云计算实验室，每天都在不断进步。

一般来说，一个简单的使用RC2的场景包括以下几个步骤：（1）用户通过Web方式访问RC2的管理界面，创建一个新的请求（Order），请求的对象必须是RC2所能提供的资源，比如虚拟机、存储等；（2）用户指定请求的服务级别协定和所需要的计算资源的数量，并提交请求；（3）管理员批准该请求，RC2会部署相应的资源，为用户提供所需的服务；（4）RC2监控用户使用资源的情况，在使用结束以后，给用户提供账单。

### 8.1.7 IBM云环境管理解决方案

云计算在我国实施，必须与我国的具体实践相结合。IBM云环境管理解决方案正是IBM中国研究院在充分了解客户需求的基础上构建的云基础设施层解决方案，也是本书作者在虚拟化和云计算领域的重要研究成果之一。针对云计算中的解决方案的规划、构建、部署、管理和优化的生命周期管理，以及各个阶段所面临的主要挑战，该解决方案提供了完备的支撑技术。该解决方案可以与IBM中国研究院的平台云和云应用结合，为各个行业提供业界领先的云服务，不仅能做到“随需应变”地满足市场需求，而且能够缩短企业制定战略到真正实施的时间，降低企业设计、采购和构

建硬件和软件平台的成本。

该解决方案重点关注如何自动化地构建和管理企业私有云，该方案既整合了IBM公司全球目前已有的虚拟化与云计算产品所提供的丰富功能，例如WebSphere CloudBurst Appliance、IBM Systems Director、Tivoli Provisioning Manager和Tivoli Monitoring Server等，又针对云环境智能构建和管理的挑战整合了很多创新性的研究成果，成为云计算解决方案生命周期管理的完整方案，如图8.5所示。该解决方案的主要功能如下。

- **集成部署与快捷上线**：在只有物理资源和网络连接的情况下，自动化构建云计算环境，支持各个层次的云计算解决方案的自动化快速上线、升级和卸载。
- **集中监控与简化管理**：通过整合的云环境管理界面，提供对云中解决方案的集中监控，简化大多数常用的解决方案管理操作，例如激活、配置和开关等。
- **性能优化与动态伸缩**：基于解决方案的性能监控信息，以及用户定义的性能优化策略和SLA，生成解决方案的性能优化方案，并动态调整分配给解决方案的各种计算资源以取得优化的性能。
- **资源优化与能源管理**：综合考虑云环境中虚拟机的资源占用情况，以及资源使用需求，通过全局的优化调度算法，在确保解决方案性能的前提下，实现云环境资源使用率的最优化。根据能耗模型，测量云环境的整体能耗及各单元能耗，通过停机、睡眠、唤醒等技术手段，实现云环境能耗最低化。

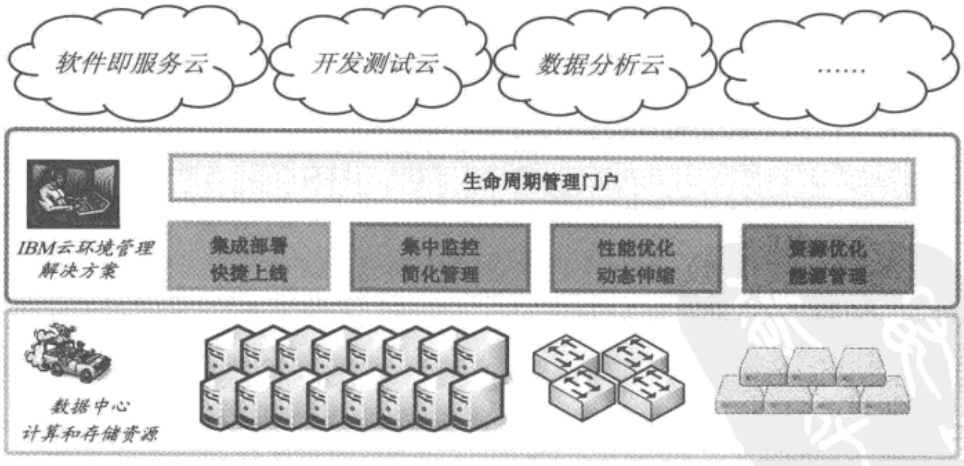



图8.5　IBM云环境管理解决方案

### 1. 构建云环境

为了适应动态变化的业务需求，企业信息部门需要提高新产品在企业私有云中的上线速度和自动化程度，从而快速响应市场变化、节约企业开销。企业私有云对于自动化部署的需求较为复杂，比如部署虚拟器件以启用新的业务系统、在虚拟机中部署软件以扩展或升级现有业务系统。因此，企业私有云需要的是端到端的自动化集成解决方案部署能力，既要支持虚拟器件和虚拟解决方案的部署，又应支持在物理和虚拟环境下的软件安装。

部署系统实现了云计算环境中混合解决方案的自动化部署，支持大规模、分布式的复杂解决方案的部署、更新及卸载，实现了从裸机环境到生产环境的自动化。为了支持应用、平台和虚拟器件等不同类型解决方案的部署，部署系统定义了标准的部署镜像格式，由解决方案的安装介质、安装逻辑脚本、元数据等组成。采用这种部署镜像格式可以将物理解决方案和虚拟解决方案进行统一打包，使每个解决方案可以由部署系统进行统一分发、统一部署，降低了管理和操作的复杂度。部署系统的公共程序接口定义了自动化部署系统的服务标准接口，目前主要包括资源预约管理服务、部署管理服务及部署镜像库管理服务，利用这些服务接口可以将部署系统同其他系统进行高效集成。

部署系统由四个核心模块构成：部署镜像模板库、资源管理模块、部署引擎及部署调度器。

部署镜像模板库负责统一管理部署系统中所有的部署镜像，根据解决方案的元数据生成镜像管理元数据，提供对部署镜像的查询、更新、上传、下载和分发等功能。在云环境中，大规模的资源请求是不可避免的，为了提高大规模并行部署的效率，部署镜像模板库在解决方案分发中采用了流传输和对等网络技术，减少了网络中数据的传输量，加快了部署进程。

资源管理模块统一管理系统的所有硬件资源，包括服务器、存储和网络资源。利用资源管理模块可以快速申请、分配及回收这些资源。

部署引擎能够解析镜像模型的元数据，分析解决方案的结构和内部组件的依赖关系，生成内部组件的部署序列，并驱动底层的部署工具对各个内部组件进行部署。自动化部署引擎是实现云计算环境下快速部署的重要手段，能够降低云计算环境的管理成本，极大缩短解决方案上线时间，提

高云计算环境的灵活性。

部署调度器负责在整个部署系统中进行解决方案部署的全局调度。在自动化部署平台中，所有的部署请求都是以预约的形式提交的。部署调度器负责将提交的预约在内存队列和数据库之间换入换出，为可以被处理的预约创建部署任务，从而启动解决方案自动化部署流程。部署调度器能够为多个预约同时创建部署任务，从而并行启动多个解决方案的部署流程实例，使多个解决方案能够根据调度结果并行执行，从而加速部署过程，缩短多个解决方案的集体上线时间。

### 2. 管理云环境

云计算解决方案被部署到云环境之后，解决方案的自动化管理变得非常关键。在云环境中，会同时运行数量可观的解决方案，对于云环境的管理员，需要有效地监控、管理这些解决方案，维护解决方案的性能，并保证整个云环境的高效利用和能源效率。图8.6是IBM云环境管理解决方案资源优化效果示例。同样，对于单个解决方案的管理员来说，在云环境中的解决方案管理工作应尽量简化，以便专注于业务本身。

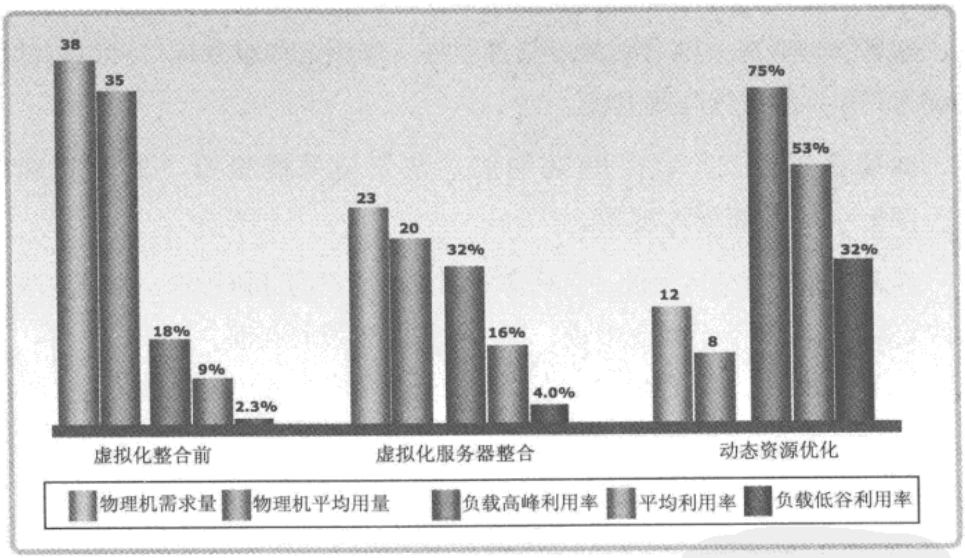



图8.6　资源优化效果示例

IBM云环境管理解决方案提供了一个应对以上挑战的自动化的、智能的云环境管理系统，其功能如下。

- **云环境资源管理**：集中展示数据中心中被管理的虚拟机、虚拟机所属的物理机及多个物理机组成的资源池的信息。根据多个解决

方案的资源调整请求，优化资源调整方案，并在数据中心级别进行全局调整。

- **云环境解决方案管理**：集中展示数据中心中被管理的所有虚拟解决方案，简要展示每个解决方案的属性信息和每个解决方案包括的虚拟器件等拓扑信息。
- **解决方案配置管理**：提供界面展示虚拟方案的所有激活和配置参数，方便用户输入定制参数和检测配置一致性。
- **解决方案状态监控**：采用图表等集中展示解决方案的监控信息，并允许用户进行定制。支持解决方案、虚拟器件或者软件的状态管理，包括启动、停止和重启等操作。
- **解决方案性能优化**：根据服务级别协议（SLA），采用服务性能检测、历史记录分析、模型预测等方法来优化解决方案性能。

解决方案元数据管理：支持系统管理员和用户对解决方案的元数据描述文件和配置脚本等进行管理。

管理系统具备三个特色：第一，统一了异构解决方案的数据访问接口，用于收集各解决方案的运行数据和性能数据，作为解决方案改进、升级的参考基础；第二，提供了对基于用户需求的虚拟机资源动态分配调整功能，确保应用、服务在保证服务质量情况下的自动化运行；第三，支持了对异构解决方案的拓扑管理、激活、启停、监控和配置等常规管理操作，降低了管理复杂度和运营成本。

管理系统提供了一个通用而可扩展的管理框架，通过解决方案开发者提供的管理元数据和脚本，可以自动化云环境管理的常用操作流程。系统提供了规范的元数据和数据接口格式，支持可视化的元数据管理功能来编辑和管理相关的元数据。元数据格式符合OVF规范，扩展了配置管理操作类型和管理脚本信息。

管理系统还支持解决方案性能优化和云环境资源管理的整体优化，管理系统在监视到解决方案性能参数的变化时，会兼顾用户定义的性能调整策略及服务级别协议，从而得到有效的性能调整方案。可能的性能调整操作类型包括对解决方案中的软件进行配置，或者请求调整虚拟机的物理资源。当需要调整虚拟机的资源时，解决方案的资源调整请求会被发送给中心管理模块，中心管理模块将根据解决方案的资源调整请求、云环境现有

的资源使用状况，得到虚拟机的资源调整方案并执行。从而实现了完全自动化、智能的动态资源优化，简化了管理员的操作，并保证了解决方案的最佳性能。

## 8.2 Amazon

Amazon公司成立于1994年，是一家业务遍布全球的电子商务企业，也是美国最大的在线零售商。在运营网上交易平台的过程中，Amazon公司积累了丰富的大规模IT基础设施管理和维护方面的经验。为了利用这些经验更好地为用户服务，Amazon公司推出了一系列云计算Web服务，本节将介绍其中最主要的几项。

### 8.2.1 概述

Amazon公司构建了一个云计算平台，并以Web服务的方式将云计算产品提供给用户，Amazon Web Services（AWS）是这些Web服务的总称。通过AWS的IT基础设施层服务和丰富的平台层服务，用户可以在Amazon公司的云计算平台上构建各种企业级应用和个人应用。用户在获得可靠的、可伸缩的、低成本的信息服务的同时，可以从复杂的数据中心管理和维护工作中解脱出来。Amazon公司的云计算真正实现了按使用付费的收费模式，AWS用户只需为自己实际所使用的资源付费，从而降低了运营成本。AWS服务包括管理计算和存储等资源的基础设施层服务和平台层服务。

AWS基础设施层服务包括Simple Storage Service（S3）、SimpleDB、简单队列服务（Simple Queue Service，SQS）和Elastic Compute Cloud（EC2）。图8.7为AWS基础设施层的基本架构，它涵盖了应用从创建、部署、运行、监控到最后卸载的整个生命周期，显示了AWS中各个Web服务之间的配合关系。用户可以将应用部署在EC2上，通过控制器启动、停止和监控应用。计费服务负责对应用的计费。应用的数据存储在SimpleDB或S3中。应用系统之间借助SQS在不同的控制器之间进行异步可靠的消息通信，从而减少各个控制器之间的依赖，使系统更为稳定，任何一个控制器的失效或者阻塞都不会影响其他模块的运行。

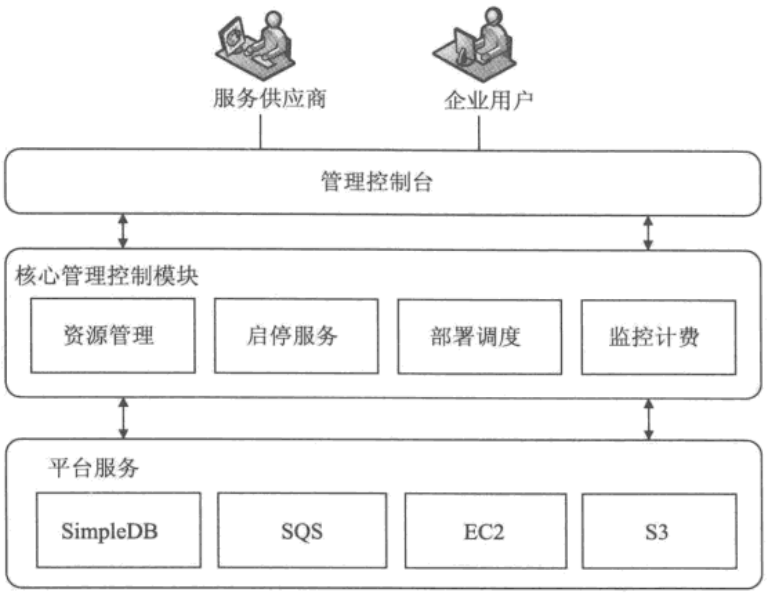



图8.7　AWS基础设施层的基本架构

AWS平台层服务包括电子商务、支付和物流等。Amazon Flexible Payments Service是专门为开发者设计的用于支付的Web服务，这个Web服务允许在任意两个实体、人或计算机之间进行支付。Amazon DevPay是用于在线计费和用户管理的Web服务，它使得开发者可以方便地对基于AWS开发的应用进行计费并对用户的账单进行管理。Amazon Fulfillment Web Service是面向商家的Web服务，使用这项服务，商家可以通过Amazon公司的物流渠道完成产品向用户的交付。Amazon Associates Web Service是一个用于电子商务的Web服务，开发者可以通过这个服务访问和使用Amazon公司的几百万个产品的数据。

### 8.2.2 Amazon S3

Amazon Simple Storage Service（S3）是云计算平台提供的可靠的网络存储服务。通过S3，个人用户可以将自己的数据放到存储云上，通过互联网访问和管理。同时，Amazon公司的其他服务也可以直接访问S3。S3由对象和存储桶（Bucket）两部分组成。对象是最基本的存储实体，包括对象数据本身、键值、描述对象的元数据及访问控制策略等信息。存储桶则是存放对象的容器，每个桶中可以存储无限数量的对象。目前存储桶本身不支持嵌套。

作为云平台上的存储服务，S3具有与本地存储不同的特点。S3采用

的按需付费方式节省了用户使用数据服务的成本。S3既可以单独使用，也可以同Amazon公司的其他服务结合使用。云平台上的应用程序可以通过REST或者SOAP接口访问S3中的数据。以REST接口为例，S3中的所有资源都有唯一的URI标识符，应用通过向指定的URI发出HTTP请求，就可以完成数据的上传、下载、更新或者删除等操作。但用户需要了解的是，S3作为一个分布式的数据存储服务，目前的版本存在着一些不足，如数据操作存在网络延迟，以及不支持文件的重命名、部分更新等。作为Web数据存储服务，S3适合存储较大的、一次写入、多次读取的数据对象，例如声音、视频、图像等媒体文件。

安全性和可靠性是云计算数据存储普遍关心的两个问题。S3采用账户认证、访问控制列表及查询字符串认证三种机制来保障数据的安全性。当用户创建AWS账户的时候，系统自动分配一对存取键ID和存取密钥，利用存取密钥对请求签名，然后在服务器端进行验证，从而完成认证。访问控制策略是S3采用的另外一种安全机制，用户利用访问控制列表设定数据（对象和存储桶）的访问权限，比如数据是公开的还是私有的等。即使在同一公司内部，相同的数据对不同的角色也有不同的视图，S3支持利用访问规则来约束数据的访问权限。通过对公司员工的角色进行权限划分，能够方便地设置数据的访问权限。如系统管理员能够看到整个公司的数据信息，部门经理能看到部门相关的数据，普通员工只能看到自己的信息。查询字符串认证方式广泛适用于以HTTP请求或者浏览器的方式对数据进行访问。

为了保证数据服务的可靠性，S3采用了冗余备份的存储机制，存放在S3中的所有数据都会在其他位置备份，保证部分数据失效不会导致应用失效。在后台，S3保证不同备份之间的一致性，将更新的数据同步到该数据的所有备份上。





### 8.2.3 Amazon SimpleDB

Amazon SimpleDB是一种支持结构化数据存储和查询操作的轻量级数据库服务。与传统的关系数据库不同，SimpleDB不需要预先设计和定义任何数据库Schema，只需定义属性和项，即可用简单的服务接口对数据进行创建、查询、更新或删除操作。

SimpleDB的存储模型分为三层：域（Domain）、项（Item）和属性（Attribute）。域是数据的容器，每个域可以包含多个项。在SimpleDB

中，用户的数据是按照域进行逻辑划分的，所以数据查询操作只能在同一个域内进行，不支持跨域的查询操作。项是由若干属性组成的数据集合，它的名字在域中是全局唯一的。项与关系数据库中表的一行类似，用户可以对项进行创建、查询、修改和删除操作。但又与表的一行有所差异，项中的数据不受固定Schema的约束，项中的属性可以包含多个值。属性是由一个或者多个文本值所组成的数据集合，在项内具有唯一的标识。在SimpleDB中，属性与关系数据库中的列类似，不同的是每个属性可以同时拥有多个字符串数值，而关系数据库的列不能拥有多个值。

SimpleDB是一种简单易用的、可靠的结构化数据管理服务，它能满足应用不断增长的需求，用户不需要购买、管理和维护自己的存储系统，是一种经济有效的数据库服务。SimpleDB提供两种服务访问方式：REST接口和SOAP接口。这两种方式都支持通过HTTP协议发出的POST或者GET请求访问SimpleDB中的数据。SimpleDB使用简单，例如数据索引是由系统自动创建并维护的，不需要程序员定义。

然而，SimpleDB毕竟是一种轻量级的数据库，与技术成熟、功能强大的关系数据库相比有些不足，比如由于数据操作是经过互联网进行的，不可避免地有较大延迟。SimpleDB不能保证所有的更新都按照用户提交的顺序执行，只能保证每个更新最终成功，因此应用通过SimpleDB获得的数据有可能不是最新的。此外，SimpleDB的存储模型是以域、项、属性为层次的树状存储结构，与关系数据库的表的二维平面结构不同，因此在一些情况下并不能将关系数据库中的应用迁移到SimpleDB上来。

### 8.2.4 Amazon SQS

Amazon Simple Queue Service（SQS）是一种用于分布式应用的组件之间数据传递的消息队列服务，这些组件可能分布在不同的计算机上，甚至是不同的网络中。利用SQS能够将分布式应用的各个组件以松耦合的方式结合起来，从而创建可靠的Web规模的分布式系统。松耦合的组件之间相对独立性强，系统中任何一个组件的失效都不会影响整个系统的运行。

消息和队列是SQS实现的核心。消息是可以存储到SQS队列中的文本数据，可以由应用通过SQS的公共访问接口执行添加、读取、删除操作。队列是消息的容器，提供了消息传递及访问控制的配置选项。SQS是一种支持并发访问的消息队列服务，它支持多个组件并发的操作队列，如向同

一个队列发送或者读取消息。消息一旦被某个组件处理，则该消息将被锁定，并且被隐藏，其他组件不能访问和操作此消息，此时队列中的其他消息仍然可以被各个组件访问。

SQS采用分布式构架实现，每一条消息都可能保存在不同的机器中，甚至保存在不同的数据中心里。这种分布式存储策略保证了系统的可靠性，同时也体现出其与中央管理队列的差异，这些差异需要分布式系统设计者和SQS使用者充分理解。首先，SQS并不严格保证消息的顺序，后送入队列的消息可能晚些时候才会可见；其次，分布式队列中有些已经被处理的消息，在一定时间内还存在于其他队列中，因此同一个消息可能会被处理多次；再次，取消息时不能确保得到所有的消息，可能只得到部分服务器中队列里的消息；最后，消息的传递可能有延迟，不能期望发出的消息马上被其他组件看到。

图8.8为一条消息的生命周期管理示例。首先，由组件1创建一条新的消息A，通过HTTP协议调用SQS服务将消息A存储到消息队列中。接着，组件2准备处理消息，它从队列中读取消息A，并将其锁定。在组件2处理的过程中，消息A仍然存在于消息队列中，只是对其他组件不可见。最后，当组件2成功处理完消息A后，SQS将消息A从队列中删除，避免这个消息被其他组件重复处理。但是，如果组件2在处理过程中失效，导致处理超时，SQS将会把消息A的状态重新设为可见，从而可以被其他组件继续处理。

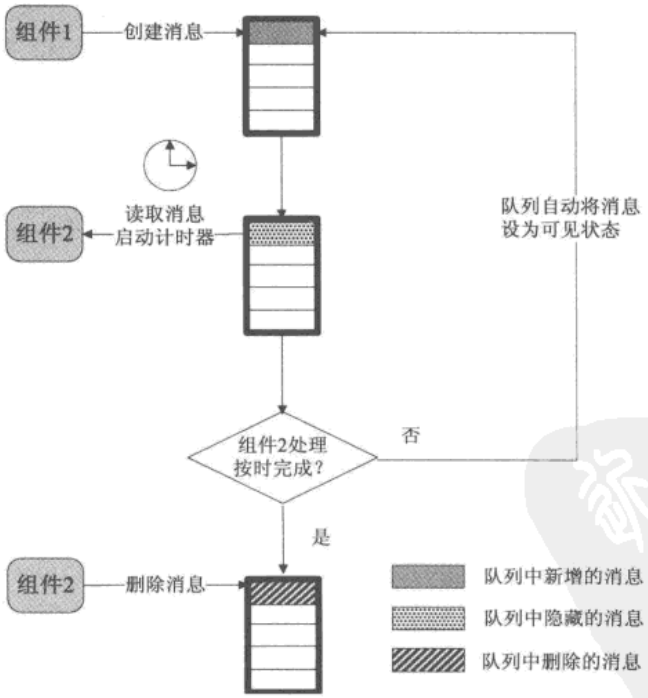



图8.8　Amazon SQS服务消息管理示例

## 8.2.5 Amazon EC2

Amazon Elastic Compute Cloud（EC2）是一种云基础设施服务。该服务基于服务器虚拟化技术，致力于为用户提供大规模的、可靠的、可伸缩的计算运营环境。通过EC2所提供的服务，用户不仅可以非常方便地申请所需要的计算资源，而且可以灵活地定制所拥有的资源，如用户拥有虚拟机的所有权限，可以根据需要定制操作系统，安装所需的软件。EC2一个诱人的特点就是用户可以根据业务的需求自由地申请或者终止资源使用，而只需为实际使用到的资源数量付费。

EC2由Amazon Machine Image（AMI）、EC2虚拟机实例和AMI运行环境组成。AMI是一个用户可定制的虚拟机镜像，是包含了用户的所有软件和配置的虚拟环境，是EC2部署的基本单位。多个AMI可以组合形成一个解决方案，例如Web服务器、应用服务器和数据库服务器可联合形成一个三层架构的Web应用。AMI被部署到EC2的运行环境后就产生了一个EC2虚拟机实例，由同一个AMI创建的所有实例都拥有相同的配置。需要注意的是，EC2虚拟机实例内部并不保存系统的状态信息，存储在实例中的信息随着它的终止而丢失。用户需要借助与Amazon的其他服务持久化用户数据，如前面提到的SimpleDB或者S3。AMI的运行环境是一个大规模的虚拟机运行环境，拥有庞大规模的物理机资源池和虚拟机运行平台，所有利用AMI镜像启动的EC2虚拟机实例都运行在该环境中。EC2运行环境为用户提供基本的访问控制服务、存储服务、网络及防火墙服务等。

通常，EC2的用户需要首先将自己的操作系统、中间件及应用程序打包在AMI虚拟机镜像文件中，然后将自己的AMI镜像上传到S3服务上，最后通过EC2的服务接口启动EC2虚拟机实例。

与传统的服务运行平台相比，EC2具有以下优势。

（1）可伸缩性：利用EC2提供的网络服务接口，应用可以根据需求动态调整计算资源，支持同时启动多达上千个虚拟机实例。

（2）节省成本：用户不需要预先为应用峰值所需的资源进行投资，也不需要雇用专门的技术人员进行管理和维护，用户可以利用EC2轻松地构建任意规模的应用运行环境。在服务的运行过程中，用户可以灵活地启、停、增、减虚拟机实例，并且只需为实际使用的资源付费。

（3）使用灵活：用户可以根据自己的需要灵活定制服务，Amazon公

司提供了多种不同的服务器配置，以及丰富的操作系统和软件组合给用户选择。用户可以利用这些组件轻松地搭建企业级的应用平台。

（4）安全可靠：EC2构建在Amazon公司的全球基础设施之上，EC2的运行实例可以被分布到全球不同的数据中心，单个节点失效或者局部区域的网络故障不会影响业务的运行。

（5）容错：Amazon公司通过提供可靠的EBS（Elastic Block Store）服务，在不同区域持久地存储和备份EC2实例，在出现故障时可以快速地恢复到之前正确的状态，对应用和数据的安全提供了有效的保障。

## 8.3 Google

Google公司拥有目前全球最大规模的搜索引擎，并在海量数据处理方面拥有先进的技术，如分布式文件系统GFS、分布式存储服务Datastore及分布式计算框架MapReduce等。2008年Google公司推出了Google App Engine（GAE）Web运行平台，使客户的业务系统能够运行在Google的全球分布式基础设施上。GAE与其他Web应用平台的不同之处在于系统的易用性、可伸缩性及成本低廉。另外，Google公司还提供了丰富的云端应用，如Gmail、Google Docs等。本节将介绍GAE平台的分布式存储服务、应用程序运行时环境、应用开发套件、Gmail和Google Docs服务。

### 8.3.1 概述

Google App Engine（GAE）平台主要包括五部分：GAE Web服务基础设施、分布式存储服务（Datastore）、应用程序运行时环境（Application Runtime Environment）、应用开发套件（SDK）和管理控制台（Admin Console），如图8.9所示。

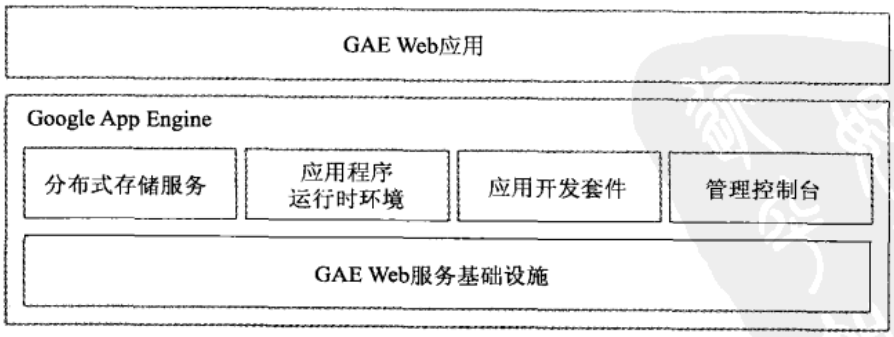



图8.9 Google App Engine系统结构

GAE Web服务基础设施提供了可伸缩的服务接口，保证了GAE对存储和网络等资源的灵活使用和管理；分布式存储服务则提供了一种基于对象的结构化数据存储服务，保证应用能够安全、可靠并且高效地执行数据管理任务；应用程序运行时环境为应用程序提供可自动伸缩的运行环境，目前应用程序运行时环境支持Java和Python两种编程语言；开发者可以在本地使用应用开发套件开发和测试Web应用，并可以在测试完成之后将应用远程部署到GAE的生产环境；通过GAE的管理控制台，用户可以查看应用的资源使用情况，查看或者更新数据库，管理应用的版本，查看应用的状态和日志等。

GAE不同于Amazon公司的EC2，EC2的目标是为了提供一个分布式的、可伸缩的、高可靠的虚拟机环境。GAE更专注于提供一个开发简单、部署方便、伸缩快捷的Web应用运行和管理平台。GAE的服务涵盖了Web应用整个生命周期的管理，包括开发、测试、部署、运行、版本管理、监控及卸载。GAE使应用开发者只需要专注核心业务逻辑的实现，而不需要关心物理资源的分配、应用请求的路由、负载均衡、资源及应用的监控和自动伸缩等任务。

### 8.3.2 分布式存储服务

GAE提供的分布式存储服务基于BigTable技术，支持结构化数据查询和更新操作，并提供事务处理功能，从而保证数据的一致性。该服务能够随着应用数据需求规模的变化而伸缩，满足应用不断变化的数据存储要求。分布式存储服务支持应用通过Java JDO/JPA接口或Python数据库标准接口访问和操作数据。与传统关系数据库相比，分布式存储服务的优势在于成本低、支持伸缩、并发性好且易管理。

在分布式存储服务的数据库中，每个实体在GAE中都包含一个全局唯一的键值。实体的键值可以由描述实体间关系的属性、实体类型、应用程序名称或者系统分配的数字实体ID组成。实体ID由实体内部的一个属性来表示，ID值可以由数据库自动生成，也可以由应用程序自己管理。实体的属性可以是简单的数据类型，如整数、浮点数、字符串、日期和二进制数据等，也可以是对其他实体的引用。多个实体可以构建成一个实体组，存储在分布式系统的相同的数据库节点中，从而提高数据创建和更新的性能。

分布式存储服务在数据操作上提供了一些高级特性。分布式存储服

务目前支持两种类型的事务操作：一种是将对实体的一组操作组成一个事务，保证单个实体的数据完整性；另一种将一组实体对象的操作组成一个事务，从而保证一组实体的数据完整性。为了支持应用对数据进行灵活的查询操作，分布式存储服务定义了专门的语言GQL，GQL的语法与SQL的语法非常相似。为了提高查询效率，GAE应用程序采用一个配置文件来定义数据的索引，在应用执行查询语句的时候，数据存储区能够直接从相应索引中获取结果。为了保证数据的一致性，分布式数据存储服务采用了乐观的并发控制（Optimistic Concurrency Control）策略。乐观并发控制策略假定大多数数据事务和其他事务不冲突，当多个应用同时访问同一数据实体时，首先将数据实体保存到本地，更新的数据只有在没有事务冲突的情况下才能直接写入数据库；如果有事务冲突，分布式数据存储服务会调用相应的冲突解决算法，或者终止事务。由于HTTP协议是无状态的协议，加锁机制在分布式存储服务的并发控制中是不可行的。因此，乐观并发控制便是一种自然的选择，不仅实现起来简单，而且减少了不必要的等待时间。



### 8.3.3 应用程序运行时环境

GAE的应用程序运行时环境是一个可伸缩的Web程序运行平台，目前能够支持Python和Java两种开发语言。

用户可以选择自己熟悉的环境支持的编程语言进行Web应用的开发。以Java为例，GAE上的Web应用程序基本遵循了JavaEE规范，开发人员可以使用Google Web Toolkit这样的Web开发框架加速开发进度和提高应用程序质量。GAE运行环境采用的是Java 6，环境包括了JavaSE Runtime Environment 6平台和库，应用可以在GAE沙盒的限制范围内使用任何JVM的字节代码或者库。为了保证GAE的性能和伸缩性，GAE对JVM进行了限制，比如在字节码中尝试打开一个套接字或者写入文件时，GAE将会抛出一个运行时异常。另外，GAE支持不同版本的应用程序同时运行，每次上传的应用都会作为一个新的版本独立地运行。

运行在GAE上的应用可以使用Google公司提供的丰富的应用服务，包括分布式数据存储服务、网址抓取、邮件、图像和Google账户等，使用Java和Python语言开发的GAE Web应用程序都能够使用这些服务。

### 8.3.4 应用开发套件

GAE为Web应用的本地开发提供了一个应用开发套件（Software Development Kit，SDK）。该SDK能够使开发人员在本地执行开发测试任务及管理和上传应用程序，其包含的Eclipse GAE插件能够极大简化在Eclipse环境中的Web应用开发和管理任务。

在开发环境中，应用可以运行在SDK提供的应用程序运行环境的安全沙盒中，这个环境可以模拟大部分API，检查到是否存在禁用模块的导入，以及系统资源的非法访问。在安全沙盒环境中，应用程序仅对操作系统拥有有限的访问权限，例如应用只能通过网址抓取服务和电子邮件服务访问互联网上的其他计算机，其他计算机也只能通过HTTP请求来访问应用程序。

当开发者进行应用的开发和测试工作时，可以利用开发套件提供的部署工具将应用程序文件和相应的配置文件上传到远程的GAE生产环境中。GAE SDK提供的Eclipse插件使得GAE应用的开发、调试和部署变得非常容易，比如在创建Web应用程序时会自动配置类路径，在开发完成后开发人员通过简单的鼠标单击就可以完成应用部署。

### 8.3.5 云端应用

Google公司的云端应用建立在其分布式的基础设施之上，能够根据用户请求的数量自动地扩展、平衡负载，并且能够通过多种有互联网接入的终端进行访问，吸引了大量的用户群。本小节着重介绍Google Docs和Gmail这两个云端应用。

Google Docs是基于Web的文字处理和电子表格程序，支持用户直接在线创建和编辑文档。Google Docs支持在线协作，团队成员可以根据授权同时在线对文档进行编辑和更新，并且能够实时看到其他成员对同一文档所做的并行修改。另外，Google Docs会自动保存用户所有的修订，使得用户对文档的修改记录一目了然并且可以根据需要恢复到之前的任何版本。同时，Google Docs集成了Google的强大的搜索能力，可以快速地对文档进行检索。

Gmail是Google的电子邮件服务，不但提供了常见的个人用户的电子

邮件服务，还提供了企业用户的电子邮件服务，使企业摆脱了开发、管理和维护邮件系统的工作，专注在能够为企业创造商业价值的业务上。Gmail不仅是有效的电子邮件工具，还集成即时消息和视频功能。用户可以通过浏览器随时了解自己的联系人的状态，同他们展开实时交流。即时消息会话内容被保存在Gmail内，用户可以像检索邮件一样对消息会话记录进行检索。除此之外，Gmail拥有强大的防病毒、过滤垃圾邮件等功能，支持移动访问，这些特点让Gmail成为极其完善的面向组织的邮件解决方案。

## 8.4 Salesforce.com

Salesforce.com公司创立于1999年，在“软件即服务”的理念指导下，该公司开发了面向企业用户的在线CRM解决方案。这种在线交付应用的服务模式免去了用户维护软、硬件设施和安装升级应用等问题，获得了很好的市场反响。

在此基础上，Salesforce.com公司推出了“平台即服务”产品Force.com。Force.com作为企业级应用的开发、发布和运营的通用平台，不再局限于某个单独的应用。该平台提供的工具和服务既可以帮助软件开发商快速开发和交付应用，又可以对应用进行有效的运营管理。下面将对Force.com平台进行详细的介绍。

### 8.4.1 概述

Force.com是平台云，它的目标是向企业用户提供云计算服务，包括按需、灵活的资源使用模式，高可靠性的服务保障，高效的开发平台及丰富的基础服务。这使得企业用户不需要再去建立数据中心，购买软、硬件设备，运营和维护数据中心的基础设施等。

Force.com向企业用户主要提供了三方面的支持：第一，直接提供在线的企业应用，比如CRM，企业用户通过简单的定制化操作就可以使用；第二，Force.com提供了一种新的编程语言Apex和集成开发环境Visualforce，能够降低应用开发的复杂度并缩短开发周期；第三，Salesforce.com公司创建了一个共享的应用资源库AppExchange，该资源库集中了企业用户和ISV在Force.com上开发的应用，并且使得应用的共享、交换及安装过程只需要通过简单的操作便可以完成，从而使Force.com的用户可以方便地把AppExchange中共享的应用集成到自己的应用中去。

Force.com提供了核心的基础服务、丰富的应用开发和管理维护服务。Force.com的基础服务为开发随需应变的应用提供了支持，其核心是多租户技术、元数据和安全架构。在基础服务之上，Force.com提供了数据库、应用开发和应用打包等服务。下面将对这些服务进行介绍。

## 8.4.2 基础服务

Force.com基础服务为上层服务和应用提供了安全、可靠的支撑环境。基础服务主要包含三个关键技术：多租户、元数据和安全架构。

在第7章我们介绍了多租户技术，它是一种共享软、硬件的技术，通过虚拟划分技术将软、硬件资源以服务的方式提供，从而可以同时支持多个客户，所有的用户都共享底层的软、硬件基础设施。在传统资源使用模式中，每个客户需要独占一套软、硬件资源，并且需要为这些资源的管理和维护花费额外的费用。采用多租户体系结构的每个客户不是独占所有的资源，而是拥有一套资源的虚拟划分。Force.com采用了多租户的体系结构，使得平台在快速部署、低风险和快速创新等方面得到了广泛认可。

元数据是Force.com的第二个关键技术。该技术简化了应用开发的复杂度。开发者不仅可以利用代码，而且可以采用元数据构建复杂的应用程序。Force.com通过元数据来描述应用的每个组件，在这个基础上，开发者可以方便地通过组合来创建更复杂的应用。采用元数据模型的另外一个好处就是，系统可以将应用和平台逻辑分开，使平台的维护和升级等操作可以和应用隔离，使底层的变化不会对上层应用造成影响。这个模型的优势已经在Force.com的平台上得到了验证，每年Force.com平台都会进行若干次主要的升级，而不会影响该平台上运行的应用。

Force.com提供了一个健壮且灵活的安全架构，能够管理用户、网络及数据。Force.com的安全架构主要包括三个方面：用户认证及授权、编程安全和平台安全框架。用户认证及授权提供了对应用、数据及逻辑访问的安全控制，保证数据和逻辑不会被未授权的用户非法访问，它主要是通过检验用户的身份及限定用户操作来实现的，如限定用户访问系统的时间，或者限定访问系统的用户IP。由于Force.com给用户提供了丰富的Web Service API及Metadata API，所以需要对这些API的调用进行安全认证，编程安全主要负责对用户调用Force.com平台的服务进行安全控制。平台安全框架包括三种粒度的安全控制：首先是系统权限，负责为用户分配Force.com平台的访问和操作

权限；其次是组件权限，负责对公司内部的不同组件的授权和管理；最后是基于记录的共享，为对象中的每个记录分配访问权限。为了保障网络和基础设施层的安全，Force.com严格遵守SysTrust SAS 70 Type Ⅱ安全标准。

### 8.4.3 数据库服务

数据库服务是Force.com平台的重要组成部分，它不仅负责应用数据的持久化，还能够通过数据对象构建相应的用户界面，方便用户对数据进行增、删、查、改。下面我们将介绍Force.com数据库服务颇具特色的三个方面：数据模型、数据操作和访问控制。

Force.com数据库服务的数据模型有两大特点。第一，数据对象持久化。在传统的关系型数据库中，数据都存储在表格中，每个表格有若干列，每个列具有固定的数据类型，不同表格之间通过外键相互关联，应用程序在读取或者写入持久化数据的时候需要将对象的属性对应在相应的列上。而Force.com数据库持久化的是数据对象，每个数据对象具有若干属性，每个属性的数据类型必须属于Force.com所规定的数据类型。第二，采用关系属性定义数据对象间的关系。传统数据库利用主键和外键来定义表格之间关联关系，而Force.com数据库通过关系属性来定义对象间的关系，并且对象间的关系只能有两种。（1）查找关系：这种关系使得用户能够从一个对象访问到另外一个对象；（2）父子关系：处于该关系中的所有子对象都需要包含关系属性，父对象的属性值是由相应子对象的数据生成的，比如某个属性值是子对象中对应属性值的最大值。

为了方便用户进行数据操作，Force.com数据库服务提供了两种交互方式：Web页面和编程接口。通过友好的Web用户界面，用户可以对存储的数据对象进行增、删、查、改和其他管理操作，从而带给用户较好的体验。另外，用户也可以使用Apex编程语言来访问数据库所提供的各种数据管理服务，Apex定义了专门的语法来帮助应用程序实现数据的查询、遍历、更新和持久化等操作。

Force.com提供了一系列的安全机制来保护用户数据的安全。在访问控制方面，提供了两种安全级别：管理安全（Administrative Security）和记录安全（Record Security）。在管理安全中，为了方便对数据进行访问控制，Force.com定义了一个类似于用户组的概念——概要（Profiles）。每个用户只能隶属于一个概要，然后对概要设定访问数据对象的增、删、查、改权

限，这些设定只能由管理员完成。记录安全提供了更细粒度的访问控制，它能精确到对数据对象某个属性的操作权限的设置。

### 8.4.4 应用开发服务

开发平台是Force.com提供的在线开发平台。通过平台提供的应用开发服务和用户界面服务，开发者可以快速地创建企业级应用。

开发者一方面可以利用Force.com提供的多租户技术的优势，包括内置的安全性、可靠性、可升级性及易用性等，另一方面可以充分利用Force.com的开发和交流平台，将发布在AppExchange上的应用服务集成到自己的项目中。利用Force.com开发平台的显著优势是开发者可以将主要精力集中在能创造商业价值的核心业务逻辑的实现上，节省硬件和软件管理、升级维护及监控等方面的成本。

针对不同类型的需求，Force.com提供了两种不同的应用开发方式。对于大多数定制功能，用户只需要通过Force.com提供的工具“单击”一些按钮就可以完成，不需要编程。另外，Force.com提供了新的编程语言Apex和完善的开发工具Visualforce来满足开发者更灵活的定制需求，并且支持分析、离线访问和移动开发。下面我们将介绍Apex和Visualforce。

Apex是为Force.com平台而设计的编程语言，它为开发者提供了一个新的构建商业应用的工具。采用Apex能够简化复杂的流程和商业逻辑，摆脱传统软件的束缚。同时，Apex无论对已有功能的定制还是对创建新的应用都具有灵活性。另外，第三方的开发者可以采用和Force.com开发团队相同的工具开发新的应用及定制已有的应用和服务。由于这些应用最终都将在Force.com平台上运行，所以开发者可以摆脱客户端应用相关问题的困扰。在Apex开发环境中，开发者可以通过界面及事件方式同用户交互，可以在服务器端操纵数据、使用信道事务（Channel Transactions）及实现流程控制。利用这些功能，开发者可以实现很多功能，比如创建个性化组件、定制或者修改已有的Salesforce.com代码、创建触发器和存储过程，以及创建和执行复杂商业应用。

Visualforce提供了简单用户界面的Apex语言的编程环境。它采用传统的模型—视图—控制器设计模式，支持数据库紧密集成，能够自动创建数据库控制器。开发者可以利用Apex实现自定义的控制器或者对已有控制器进行扩展。Visualforce包含基于标签的标记性语言和数十种内置组件，有

足够的灵活性来支持开发者创建自定义的组件和界面。

### 8.4.5 应用打包服务

Force.com提供的应用打包（Packaging）服务能够将开发者创建的应用发布出去。Force.com所定义的包（Package）是代码、功能组件或者应用的集合，它向外界提供的可能是一个单一的功能组件，也可能是一系列应用组成的整体解决方案。

Force.com有两种格式的包：非受控包（Unmanaged Package）和受控包（Managed Package）。非受控包适合于只需要发布一次的组件和应用，它类似于模板，一旦创建完成，就可以生成实例给用户使用，因此非受控包适合用于共享应用模板和代码示例。相对于非受控包，受控包提供了知识产权方面的保护，因为包中许多功能组件的源代码对外界都是不可见的。不仅如此，受控包的开发者还能够对包进行升级。受控包适合用于发布收费的应用，并对发布的应用提供许可证支持。

在Force.com平台上，通过应用打包服务打包并发布应用的步骤大致分为三步：创建、上传和注册。下面我们将具体介绍这三个步骤。

在创建阶段，开发者需要将自己的代码、功能组件或者应用进行打包。不过，非受控包和受控包的创建过程有所不同。创建非受控包的流程比较简单，而且所有身份的开发者都可以创建。首先，开发者在Force.com提供的个人页面上创建一个空包，并给该包命名，然后逐一向该包里添加内容项（Item），最后保存。Force.com定义了很多内容项的类型，比如Apex类、Apex触发器、文档或控件，在添加内容项时要先选择相应的类型。对于受控包的创建，Force.com提出了严格的要求：第一，开发者必须具有Developer Edition的身份；第二，为了防止和其他受控包冲突，开发者必须在Force.com注册命名空间前缀（Namespace Prefix）。在给受控包添加完内容项之后，开发者需要注册命名空间前缀，并且制定刚才创建的受控包，保存以后，Force.com会提示受控包创建成功。

上传过程包含简单的三步操作：第一，开发者进入Force.com提供的个人页面上选择所要上传的包；第二，定义这次上传包的版本和添加相应的描述；第三，上传完成以后，Force.com会返回一个该应用的URL链接，开发者可以将该链接发布给其他用户。对于受控包，在第二步的提示中还会

要求开发者选择受控包是测试版（Managed-Beta）还是正式版（Managed-Release）。

通过注册，开发者可以将自己的应用发布到AppExchange中和其他用户分享。根据共享的范围不同，分为私有包（Private Packages）和公有包（Public Packages）。私有包的应用在特定的群体和社区内共享，而公有包的应用对AppExchange上所有的用户都是可见的。上传以后，开发者在AppExchange页面上通过创建或修改包的某些属性对包进行注册，不过只有走完AppExchange的审核流程以后，才能成为公有包。

## 8.5 Microsoft

Microsoft公司的软件产品覆盖操作系统、软件开发平台、数据库、办公软件等领域。面对云计算这个可能改变IT产业格局的新机遇，该公司于2008年10月正式推出了云计算产品Windows Azure平台。Windows Azure平台是运行在Microsoft数据中心，为互联网用户提供服务的一组云计算技术的集合。

### 8.5.1 概述

Windows Azure平台由Windows Azure及一组平台层服务构成，如图8.10所示。Windows Azure平台的基础设施层组件是Windows Azure，它作为云平台的操作系统被安装在提供云服务的数据中心的每台服务器上。Windows Azure管理着数据中心所有的服务器、存储和网络等资源。Windows Azure平台给云应用层提供的平台层服务包括：（1）.NET服务，为云应用和本地应用的开发提供了支持；（2）SQL Azure，方便用户以服务的方式访问和使用云中的Microsoft SQL Server数据库；（3）Live服务，通过该服务开发者可以访问Microsoft Live应用和其他应用的数据，并且能够在不同的设备之间同步应用数据、查找和安装软件等；（4）SharePoint服务，提供了一个可伸缩和可管理的平台，使用户能够协作开发基于Web的业务应用程序；（5）Dynamic CRM服务，为软件开发者提供了一套构建复杂商业应用的基础服务。由于采用了如SOAP和REST等标准的Web通信协议，这些服务能够很好地和用户的应用及其他云平台进行集成。下面将介绍Windows Azure平台中的核心组件。

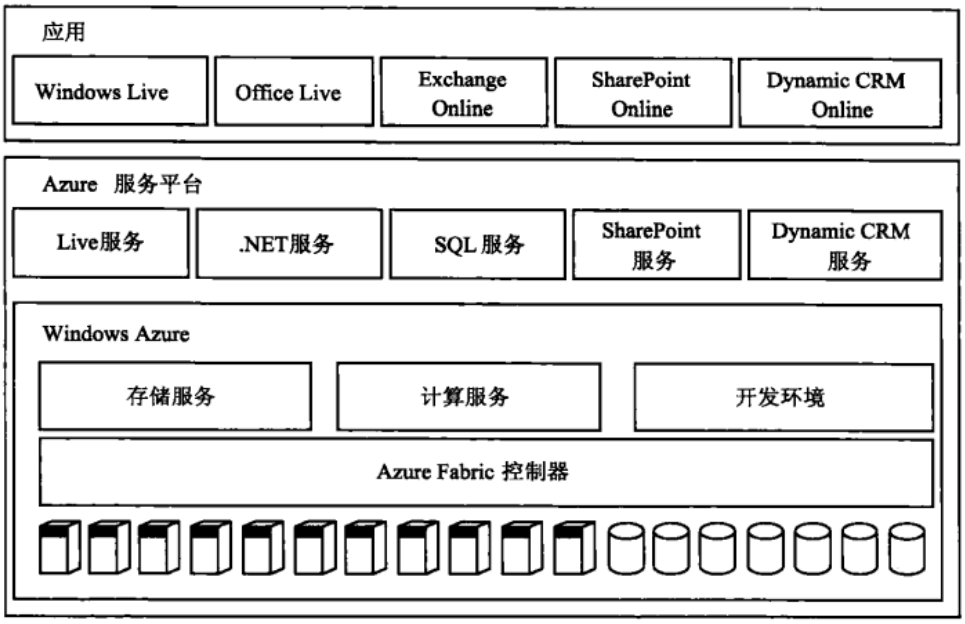



图8.10 Windows Azure平台

## 8.5.2 Microsoft Windows Azure

Windows Azure是Microsoft云平台上的操作系统，运行在Microsoft的数据中心中。操作系统作为基础设施的调度和管理软件，为构建高效、可靠、可伸缩的云计算平台起着重要的作用。Windows Azure由四部分组成：Windows Azure Fabric、存储服务、计算服务和云应用开发环境。Fabric类似虚拟化中虚拟机监视器的资源管理功能，它能够将数据中心的服务器、网络和存储等资源组成一个逻辑的资源池，统一管理云中资源。存储服务、计算服务和开发环境是Windows Azure对外提供的服务，三个服务相互独立，开发者可以根据需要选择自己需要的服务。采用存储服务应用可以从Microsoft公司获得可靠的数据存取及管理服务。计算服务为应用提供了一个可靠、可伸缩的运行环境。而通过应用开发环境，用户可以快速、高效地开发基于Windows Azure平台的应用。下面分别介绍Windows Azure中的几个关键模块：Windows Azure Fabric、存储服务、计算服务和开发环境。

Windows Azure Fabric负责Windows Azure平台所管理的云中各种资源，包括存储设备、服务器、交换机和负载均衡设备的分配、部署、监控、管理、维护和回收。Windows Azure Fabric由其管理的大量的IT设备及相应的管理软件Fabric控制器组成。应用被上传到Windows Azure平台后，Fabric控制器通过读取应用的配置文件，为应用创建虚拟机，并根据硬件资

源优化调度策略分配物理资源。每个应用包括至少两个配置文件：服务定义和服务配置。这两个文件描述了应用所需的账户信息、认证信息、存储配置信息及其他资源的需求信息，如需要多少虚拟机实例来运行Web Role服务器，以及需要多少虚拟机实例运行Worker Role服务器等。应用部署完成后，Windows Azure Fabric便立即开始监控应用的状态，以保证应用可靠、稳定地运行。为了使Fabric能够实时地获取应用的状态，所有的虚拟机都预先安装了Fabric代理，然后通过这个代理与Fabric控制器进行通信，从而获取应用的实时情况。当检测到虚拟机故障时，Fabric控制器会启动新的包含同样服务的虚拟机实例对外提供服务。同样，如果检测到物理机故障导致无法提供服务，Fabric会立即将运行在这台物理机上的所有的虚拟机实时迁移到其他物理机上。

Windows Azure目前提供了针对三种数据结构的存储服务以满足应用的不同需求，这三种数据结构是Blob、表（Table）和队列（Queue）。Blob存储服务能够支持用户存储数据量大的数据集合。在Blob存储服务中，每个用户的数据都是按照层次结构存储在和自己账户关联的逻辑存储空间中。每个Blob存储服务用户的数据首先以容器（Container）的粒度进行划分，一个数据容器通常代表了用户对数据的一个分类；每个容器中又可以存储一个或者多个Blob，一个Blob可能达到几个甚至几十个GB，为了提高Blob的数据传输效率，每个Blob又可以分为多个Block。在Blob的传输过程中，如果发生数据丢失，只需要重传对应的Block而不是整个Blob。表存储服务用于满足应用存储结构化数据的需求。表由实体（Entity）和属性组成，同样采用层次化的存储结构。每个表包含若干实体，每个实体又由一组属性组成。实体的属性可以是不同的类型，如整数类型、字符串、布尔类型和日期类型等。每个表可以存储数十亿计的实体，表的大小可以达到TB级别。队列存储服务是Windows Azure提供的第三种存储服务，它用于为不同的应用之间或者应用的不同模块之间提供可靠的、持久化的消息服务。

Windows Azure的存储服务还提供了很多特性来进一步保证数据的可靠性、访问效率和可扩展性。为了保证数据的可靠性，Windows Azure存储的每份数据都会在至少三个物理服务器上进行冗余备份，当某份数据失效的时候，应用可以通过访问备份继续访问数据，同一数据的不同备份之间的一致性是由系统自动维护的。为了提高数据访问的性能，Windows Azure将数据表的数据内容进行分割，分别存储到不同的节点上，并采用并行机制进行访问，从而提高数据的访问效率。为了提高存储的可扩展性，方便不同类型应

用的访问需求，Windows Azure的存储服务支持通过RESTful方式进行访问，以便这些存储服务不仅能够被Azure应用使用，也能更容易地被其他技术平台的应用进行集成。

Windows Azure提供了一个可伸缩的计算环境。由于虚拟机可以容易地实现资源伸缩调整，因此Windows Azure采用了虚拟机作为Windows Azure平台上应用的运行环境。每个虚拟机中运行的是Microsoft的操作系统Windows Server 2008，而虚拟机管理软件是基于Microsoft公司的Hyper-V实现的。Microsoft公司目前提供的Windows Azure版本支持两种虚拟机类型：一种是Web Role，负责接收客户端的HTTP请求；另外一种是Worker Role，负责从Web Role接收输入和执行计算，并将计算结果返回给Web Role或者写到指定的存储位置。为了支持应用的伸缩，Web Role被限定为无状态的，从而使得应用能够在负载较重的时候，非常方便地增加Web Role实例的数量，提高应用支持的并发访问量；当应用负载变小时，也可以方便地减少Web Role实例而不会对应用的运行产生影响。

Windows Azure应用的开发也比较方便。目前Azure提供了集成的开发环境，如Visual Studio或者添加了Windows Azure开发插件的Eclipse，通过这些工具，开发者可以快速地构建Windows Azure应用程序。熟悉Visual Studio的开发者会发现，开发一个Windows Azure上的应用和开发其他熟悉的项目是非常类似的，都需要选择熟悉的语言创建一个新的项目，实现应用的逻辑、调试，以及最后打包发布。不同的是，基于Windows Azure的应用是分布式的（可能包括多个Web Role实例和Workers Role实例），所以调试的方式有所不同。目前，系统支持通过日志的方式调试，通过调用系统API来记录应用的状态信息。除此之外，开发者还需要理解，本地环境是一个模拟的环境，当应用开发完成后，需要对服务的配置信息进行修改，如将存储账户和地址等信息替换成生产环境的信息。修改完成后，应用才能进行打包，然后发布到Windows Azure平台上，对外提供服务。

### 8.5.3 Microsoft .NET服务

.NET服务是一个基于Web的服务，它是Microsoft公司对传统单机上的.NET框架的扩展，目标是为用户提供基于标准网络协议的Web应用开发平台，并通过对常规操作及底层细节的封装，简化用户的开发工作，使用户更多地关注于应用的功能和业务流程。.NET服务目前提供的核心模

块有两个：访问控制服务（Access Control Service）和服务总线（Service Bus）。还有更多的模块即将推出。

.NET访问控制服务为Web应用程序提供了用户身份认证和授权的功能，使应用可以定制访问资源的策略。访问控制服务主要有以下优点：（1）提供了SOAP和REST接口，从而可以灵活地与其他云应用或传统的身份机制进行集成，例如企业名录、Windows Live ID等；（2）采用了基于规则的访问控制（Rule Based Access Control）策略，通过不同的规则可以组合出复杂的访问控制策略，满足各种各样的访问控制需求；（3）提供了访问控制策略映射的功能，使得不同的认证和授权服务能够更好地协同工作。

服务总线类似于SOA架构中的企业服务总线（Enterprise Service Bus，ESB），因此熟悉SOA的用户可以很快地学习并使用。服务总线提供了服务的注册、查找和访问功能。由于企业内部不同的应用可能会运行在不同的机器上，因此服务总线还要提供网络地址转换及穿透防火墙的功能。服务总线主要有以下优点：（1）Web服务管理简单高效，通过服务总线统一管理企业内部的Web服务，可以简化服务访问地址和防火墙策略的管理；（2）便于Web服务的共享，用户可以通过服务总线非常容易地查询Web服务的信息，并根据服务总线提供的信息对Web服务进行访问；（3）高安全性，相比于直接将提供服务的服务器地址暴露在互联网上，服务总线通过与访问控制服务结合，隐藏服务器的真实地址，从而降低Web服务可能受到的威胁。

### 8.5.4 Microsoft SQL Azure

SQL Azure服务提供了一个云环境的数据管理系统，它包含了一组针对结构化、半结构化及非结构化数据的云应用数据管理技术，目的是为云应用提供一种可靠的、可伸缩的、高效的、可以通过互联网访问的数据服务，具体功能包括数据存储、数据查询、数据分析及报表等。

用户使用SQL Azure的方式和使用传统的SQL Server环境类似，用户通过已有的SQL Server客户端进行访问，也可以使用ADO.NET约定的数据访问方式进行访问。当然，SQL数据服务也有不同于传统SQL Server的地方，如不支持CLR（Common Language Runtime）、空间数据（Spatial Data）及一部分系统管理功能（如启动、停止SQL Server）。

SQL Azure服务还能够为用户带来很多传统数据管理系统不具备的好

处。首先，由于数据放置在云中，数据的常规管理都由云中的管理系统完成，因而用户可以摆脱繁重的数据库管理和维护的工作，无需对数据库进行定期备份，也不再需要定期为数据库打补丁。其次，云环境为用户提供了统一的数据访问接口，用户不需要关心数据的具体位置。在当前版本的SQL Azure服务中，每个数据库大小的上限在5GB到10GB之间，如果应用的数据小于这个限制，则可以保存在单个数据库中，否则系统会创建多个数据库，将应用数据划分在不同的数据库分别存放。在传统情况下，应用不仅需要知道所要访问的数据库，而且还需要知道每个数据库中的数据划分信息。而在SQL Azure服务中，系统会封装下层多个数据库的复杂操作，将用户提交的数据操作分发到各个数据库上执行，然后对执行结果进行合并，再返回给用户。再次，采用SQL Azure服务的应用能获取比传统单个数据库更健壮的服务。与Windows Azure数据服务类似，SQL Azure服务的每份数据都会在不同的地方进行备份。当一份数据失效时，可以从其他备份进行恢复。同时，SQL Azure服务会保证多个备份中数据的一致性，如果对数据库的更新操作返回成功信息，则意味着所有备份都已经成功进行了更新。

SQL Azure服务作为一种简单、有效、低成本的数据管理服务，为云应用提供了具备良好扩展性、可控性及可靠性的数据管理服务，它不仅能降低企业的成本，还能支持灵活的访问方式，这些都成为SQL Azure服务吸引企业的亮点。随着云计算技术的不断发展，新的需求不断涌现，SQL Azure服务将会不断丰富，从而解决云计算环境中更多面向数据处理的问题。





### 8.5.5 Microsoft Live服务

Live服务提供了对Microsoft公司的庞大的用户群数据及应用资源的管理服务。为了方便开发者基于这些数据开发个性的应用，Live服务封装了丰富的服务给开发者使用。通过使用Live服务，开发者可以方便地开发自己的社交网络软件应用，或者组装现有的应用模块。

Live服务提供了在互联网应用之间共享数据的框架和机制。Live服务的核心组件是Live框架。利用Live框架，开发者不仅可以访问Microsoft Live服务的数据，而且可以在不同的设备之间利用Live Mesh进行数据同步。Live服务主要提供两类数据的共享，一类是公有的可以被任何人访问的数据，如地图信息；另一类是含个人隐私的数据，如用户个人资料、联系人信息等，这些数

据只能提供给授权用户使用。

Live服务为资源分配了唯一的URI，应用可以利用HTTP协议发出REST请求访问这些资源。URI是用来标识资源的信息，所以如何确定URI显得尤为重要。Live框架定义了一个资源模型来统一描述和命名Live服务数据。资源模型定义了基本的资源类型、资源之间的关系及一致的URI命名规则。为了满足应用的个性化需求，资源模型允许添加用户自定义的资源类型，这样不仅能够方便应用发现和访问Live服务数据，而且方便开发者管理数据。因为数据的访问权限是由开发者控制的，所以Live服务能够有效保障数据的安全性。

为了开发跨不同设备平台的应用或者服务，最佳的方法是采用Live Mesh。Live Mesh负责在不同的设备之间同步、共享、存储和访问文件或文件夹。通过将多个不同的设备添加到一个Mesh平台系统中，Live操作环境（Live Operating Environment）能够自动地同步所有设备之间的数据。Mesh平台系统中的每个设备都是Master节点，这意味着数据的更新操作可以从任何一个设备触发。Live Mesh平台不仅能同步云应用之间的数据，而且能同步云端和本地的数据。

Live服务在Windows Azure平台产生之前就已经存在了，使用Live服务的最著名应用是Windows Live，其中集成了Live Messenger、Hotmail、Live Writer等常用的基于Web的应用。Live Messenger和Live Mail这两个应用可以共享Live用户的资料和联系人信息；Live Map利用共享的地图信息提供了导航功能；Live Search利用海量的网页信息提供信息检索功能。

## 8.6 小结

本章对云计算业界五家主要厂商的基本情况和主要产品进行了介绍。

IBM公司作为云计算领域的倡导者，不仅参与了云计算主要标准的制定工作，还提供了非常全面的云计算产品，这些产品涉及云架构的基础设施层、平台层和应用层。我们重点介绍了IBM公司部分云计算产品和解决方案，包括管理虚拟化基础设施资源的Ensembles、自动化云服务管理的TSAM、专注于SOA云环境管理的WCA解决方案、协作办公云服务应用LotusLive、提供云计算自助服务交付的RC2解决方案，以及与我国实际情况紧密结合的IBM云环境管理解决方案。

Amazon公司最为人所知的是它的在线电子商务业务，在云计算领域

Amazon也拥有非常成功的基础设施云AWS，我们对AWS所提供的一些典型服务做了详细介绍：针对存储资源的Amazon S3服务、针对数据库的Amazon SimpleDB服务、针对消息队列的Amazon SQS服务，以及提供计算资源的Amazon EC2服务。

Google公司在云计算领域的主要贡献是GAE（Google App Engine）。GAE作为云计算的平台层，为其上应用的开发提供了非常便利的环境。我们介绍了GAE的核心服务：分布式存储服务、应用程序运行时环境和应用开发套件。另外，Google公司也推出了一些云应用，比如著名的在线电子文档编辑器Google Docs和在线邮件服务Gmail。

Salesforce.com公司在初期为企业用户提供在线的Salesforce CRM解决方案，在业界引起了极大的反响，之后它又推出了基于云计算的应用开发平台Force.com。我们详细介绍了Force.com平台的基本情况，以及它提供的核心服务，比如数据库服务、整合服务等。

Microsoft公司也凭借最近推出的Windows Azure云计算平台在云计算业界占有了一席之地。Windows Azure平台由作为基础设施层的Windows Azure及一组平台层服务构成。Windows Azure作为基础设施层，管理云计算环境下的服务器、网络和存储等资源。我们所介绍的平台层服务包含了与软件开发有关的.NET服务、与数据管理有关的SQL Azure服务，以及与云应用开发有关的Live服务。

CHAPTER 附录A

# 附录A 超级计算机排名

如表A.1所示为2009年6月超级计算机500强前10名。

表A.1 2009年6月超级计算机500强前10名

| 排名 | 安装地点<br>计算机名 | 公司<br>年代 | 总核数<br>耗电量 | 实际峰值<br>理论峰值 | 配置 |
|---|---|---|---|---|---|
| 1 | DOE/NNSA/LANL<br>United States | IBM | 129 600 | 1105.00 | BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8GHz, Voltaire Infiniband / |
| | Roadrunner | 2008 | 2483.47 | 1456.70 | |
| 2 | Oak Ridge National LaboratoryUnited States | Cray Inc. | 150 152 | 1059.00 | Jaguar-Cray XT5 QC 2.3 GHz / 2008 |
| | Jaguar | 2008 | 6950.60 | 1381.40 | |
| 3 | Forschungszentrum Juelich (FZJ)Germany | IBM | 294 912 | 825.50 | Blue Gene / P solution |
| | JUGENE | 2009 | 2268.00 | 1002.70 | |
| 4 | NASA/Ames Research Center/NASUnited States | SGI | 51200 | 487.01 | SGI Altix ICE 8200EX, Xeon QC 3.0/2.66 GHz |
| | Pleiades | 2008 | 2090.00 | 608.83 | |
| 5 | DOE/NNSA/LLNL<br>United States | IBM | 212 992 | 478.20 | eServer Blue Gene Solution |
| | BlueGene/L | 2007 | 2329.60 | 596.38 | |
| 6 | National Institute for Computational Sciences/ University of Tennessee<br>United States | Cray Inc. | 66 000 | 463.30 | Cray XT5 QC 2.3 GHz |
| | Kraken XT5 | 2008 | | 607.20 | |
| 7 | Argonne National Laboratory<br>United States | IBM | 163 840 | 458.61 | Blue Gene/P Solution |
| | Blue Gene | 2007 | 1260.00 | 557.06 | |

续表

| 排名 | 安装地点<br>计算机名 | 公司<br>年代 | 总核数<br>耗电量 | 实际峰值<br>理论峰值 | 配置 |
|---|---|---|---|---|---|
| 8 | Texas Advanced Computing Center/Univ. of Texas<br>United States | SUN | 62 976 | 433.20 | SunBlade x6420, Opteron QC 2.3 Ghz, Infiniband |
| | Ranger | 2008 | 2000.00 | 579.38 | |
| 9 | DOE/NNSA/LLNL<br>United States | IBM | 147 456 | 415.70 | Blue Gene/P Solution |
| | Dawn | 2009 | 1134.00 | 501.35 | |
| 10 | Forschungszentrum Juelich (FZJ)Germany | Bull SA | 26 304 | 274.80 | Sun Constellation, NovaScale R422-E2, Intel Xeon X5570, 2.93 GHz, Sun M9/Mellanox QDR Infiniband/Partec Parastation |
| | JUROPA | 2009 | 1549.00 | 308.28 | |

数据来源：http://www.top500.org/lists/2009/06

注：实际峰值和理论峰值的单位是 TFlops；耗电量数据为整个系统的耗电量，单位是千瓦。

如表A.2所示为超级计算机绿色500强前10名。

表A.2　超级计算机绿色500强前10名

| 排名 | 运算速度能耗比 (MFLOPS/Watt) | 安装地点 | TOP 500 排名 |
|---|---|---|---|
| | 总耗电量 (KW) | 计算机配置 | |
| 1 | 536.24 | Interdisciplinary Centre for Mathematical and Computational Modeling, University of Warsaw | 422 |
| | 34.63 | BladeCenter QS22 Cluster, PowerXCell 8i 4.0 Ghz, Infiniband | |
| 2 | 458.33 | DOE/NNSA/LANL | 61 |
| | 138.00 | BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Infiniband | |
| 2 | 458.33 | IBM Poughkeepsie Benchmarking Center | 62 |
| | 138.00 | BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Infiniband | |
| 4 | 444.94 | DOE/NNSA/LANL | 1 |
| | 2483.47 | BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Voltaire Infiniband | |
| 5 | 428.91 | National Astronomical Observatory of Japan | 277 |
| | 51.20 | GRAPE-DR accelerator Cluster, Infiniband | |
| 6 | 371.67 | ASTRON/University Groningen | 124 |
| | 94.50 | Blue Gene/P Solution | |
| 7 | 371.67 | IBM - Rochester | 84 |
| | 126.00 | Blue Gene/P Solution | |
| 7 | 371.67 | IBM Thomas J. Watson Research Center | 85 |
| | 126.00 | Blue Gene/P Solution | |
| 7 | 371.67 | Max-Planck-Gesellschaft MPI/IPP | 86 |
| | 126.00 | Blue Gene/P Solution | |
| 7 | 371.67 | Bulgarian State Agency for Information Technology and Communications (SAITC) | 245 |
| | 63.00 | Blue Gene/P Solution | |

数据来源：http://www.green500.org/lists/2009/06/list.php

# 参考资料


[1] Wikipedia.Data Center.http://en.wikipedia.org/wiki/Data_center

[2] TR-42.1. TIA-942 Datacenter Standard overview, April 2005. http://www.adc.com/us/en/Library/Literature/102264AE.pdf

[3] Wikipedia. Server farm. http://en.wikipedia.org/wiki/Server_farm

[4] Rob Snevely. Enterprise Data Center Design and Methodology. Prentice Hall, Feb 2002.

[5] IBM. IBM服务管理. http://www-01.ibm.com/software/cn/tivoli/solution/it-service-management/

[6] 博恩. 基于ITIL的IT服务管理基础篇.第一版.北京：清华大学出版社，2007.

[7] IBM. The New Enterprise Data Center Technical White Paper. May, 2008.

[8] Amit Singh. An Introduction to Virtualization. January 2004. http://www.kernelthread.com/publications/virtualization/

[9] Martin F. Maldonado. 虚拟化概述：模式的观点. July 2006. http://www.ibm.com/developerworks/cn/grid/gr-virt/

[10] 晓凡. 服务器虚拟化、网络虚拟化及存储虚拟化释义. Feb 2008. http://cio.ctocio.com.cn/pinglun/385/7810885.shtml

[11] IBM. 追根溯源话虚拟. IBM虚拟技术大会. http://www-900.ibm.com/cn/itmanager/optimizeit/virtualizationworld/xpvw_01.shtml

[12] Wikipedia. 桌面虚拟化. http://en.wikipedia.org/wiki/Desktop_virtualization

[13] 陈翔.“化零为整”的新型部署模式：终端虚拟化服务. http://media.ccidnet.com/art/2619/20080819/1548489_1.html

[14] VMWare. Virtualization Basics. http://www.vmware.com/technology/virtualization.html

[15] 计世网. 五大虚拟化热门技术：CPU虚拟化居首. http://www.enet.com.cn/article/2008/1202/A20081202397144.shtml

[16] IBM developer Works. Virtual Linux, An overview of virtualization methods, archite-cture, and implementations. http://www.ibm.com/developerworks/library/l-linuxvirt/

[17] Virtualization for Data Centers of Today & Tomorrow. IBM

Corporation

[18] David Davis. Server Virtualization. Network Virtualization & Storage Virtualization Explained. 2009. http://www.petri.co.il/server-virtualization-network-virtualization-storage-virtualization.html

[19] VMWare. Understanding Full Virtualization, Paravirtualization, and Hardware Assist Virtualization. http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf

[20] Constantine Sapuntzakis, David Brumley, Ramesh Chandra, Nickolai Zeldovich, Jim Chow, Monica S. Lam, and Mendel Rosenblum. Virtual Appliances for Deploying and Maintaining Software. LISA 2003.

[21] Ruth Willenborg. Virtual Appliances Panacea or Problems. IBM developerWorks, 2007. http://www.ibm.com/developerworks/websphere/techjournal/0710_col_willenborg/0710_col_willenborg.html

[22] Hao Yu, Jose E. Moreira, Parijat Dube, I-hsin Chung, Li Zhang. Performance Studies of a WebSphere Application, Trade, in Scale-out and Scale-up Environments. IPD-PS, 2007.

[23] 王庆波，金滢，陈滢.虚拟器件技术及应用.中国计算机学会通讯.2009.6，5（6）

[24] DMTF Open Virtualization Format Specification v1.0.0, Feb. 2rd, 2009. http://www.dmtf.org/standards/published_documents/DSP0243_1.0.0.pdf

[25] Steve Schmidt, Mike Gering, Andrew Freed. Building Virtual Appliances using OVF Toolkit. IBM developerWorks, Jun. 2009.

[26] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christia-n Limpach, Ian Pratt, Andrew Warfield. Live Migration of Virtual Machines. NSDI2005.

[27] Marvin McNett, Diwaker Gupta, Amin Vahdat, and Geoffrey M. Voelker. Usher: An Extensible Framework for Managing Clusters of Virtual Machines. the Proceedings of the 21st Large Installation System Administration Conference (LISA '07). 167-181 http://www.usenix.org/event/lisa07/tech/full_papers/mcnett/mcnett_html

[28] Kyrre M Begnum. Managing Large Networks of Virtual Machines. the Proceedings of LISA '06: 20th Large Installation System Administration Conference. 205-214 https://www.usenix.org/events/lisa06/tech/full_papers/begnum/begnum_html

[29] L Grit, D Irwin, J Aydan Chase. Virtual Machine Hosting for Networked Clusters: Building the Foundations for "Autonomic" Orchestration. VTDC 2006.

[30] Brendan Cully, Geoffrey Lefebvre, Dutch Meyer, Mike Feeley, Norm Hutchinson, and Andrew Warfield. Remus: High Availability via Asynchronous Virtual Machine Re-plication. NSDI 2008.

[31] Le He, Shawn Smith, Ruth Willenborg, Qingbo Wang. Automating deployment and activation of virtual images. IBM developerWorks, 2007.

[32] http://www.ibm.com.

[33] IBM Mainframe. http://www-03.ibm.com/systems/z/?cm_re=masthead-_-products-_-sys-zseries.

[34] IBM System p. http://www-03.ibm.com/systems/p/?cm_re=masthead-_-products-_-sys-pseries.

[35] IBM Power Systems. http://www-03.ibm.com/systems/power/?cm_re=masthead-_-products-_-sys-power.

[36] IBM PowerVM. http://www-03.ibm.com/systems/power/software/virtualization/index.html.

[37] IBM Systems Director. http://www-03.ibm.com/systems/management/director/.

[38] IBM Tivoli Application Dependency Discovery Manager. http://www-01.ibm.com/software/tivoli/products/taddm/.

[39] IBM Tivoli Change and Configuration Management Database. http://www-01.ibm.com/software/tivoli/products/ccmdb/.

[40] IBM Tivoli Monitoring. http://www-01.ibm.com/software/tivoli/products/monitor

[41] VMware official web site. http://www.vmware.com/

[42] VMware Infrastructure3. http://www.vmware.com/products/vi/

[43] VMware server. http://www.vmware.com/products/server/

[44] VMware ESXi. http://www.vmware.com/products/esxi/
[45] VMware vCenter server. http://www.vmware.com/products/vi/vc/
[46] VMware vCenter Site Recovery Manager. http://www.vmware.com/products/srm/
[47] VMware vCenter Lab Manager. http://www.vmware.com/products/labmanager/
[48] VMware vCenter Lifecycle Manager. http://www.vmware.com/products/lcm/
[49] VMware vCenter Stage Manager. http://www.vmware.com/products/sm/faq.html
[50] VMware vCenter Converter. http://www.vmware.com/products/converter/
[51] VMware vCenter AppSpeed. http://www.vmware.com/products/vcenter-appspeed/
[52] VMware View. http://www.vmware.com/products/view/
[53] VMware Workstation. http://www.vmware.com/products/ws/
[54] VMware Fusion. http://www.vmware.com/products/fusion/
[55] VMware ThinApp. http://www.vmware.com/products/thinapp/
[56] VMware ACE. http://www.vmware.com/products/ace/
[57] Xen/Citrix official site. http://www.citrix.com/
[58] Citrix delivery center. http://citrix.com/English/ps2/products/product.asp?contentID=683711
[59] Microsoft Desktop virtualization. http://www.microsoft.com/virtualization/products/desktop/default.mspx
[60] Microsoft Server Virtualization . http://www.microsoft.com/virtualization/products/server/default.mspx
[61] Microsoft Application Virtualizaiont. http://www.microsoft.com/virtualization/products/application/default.mspx
[62] Microsoft Virtualization Management. http://www.microsoft.com/virtualization/products/management/default.mspx
[63] Wikipedia.com. Cloud computing. http://en.wikipedia.org/wiki/Cloud_computing
[64] Searchcloudcomputing.com. What is cloud computing. http://

searchcloudcomputing.techtarget.com/sDefinition/0,,sid201_gci1287881,00.html#

[65] Microsoft. Cloud computing definition. http://www.microsoft.com/china/CRD/en/innoforum/innoforum_14.mspx

[66] Salesforce. Cloud computing definition. http://www.salesforce.com/cloudcomputing/

[67] BusinessWeek. Google and the Wisdom of Clouds. http://www.businessweek.com/magazine/content/07_52/b4064048925836.htm

[68] BusinessWeek. Amazon and Cloud Computing. http://www.eweek.com/c/a/Cloud-Computing/Amazon-and-Cloud-Computing.

[69] Michael Armbrust, et al. Above the clouds: A Berkeley View of Cloud Computing. http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf.

[70] Hagit Attiya and Jennifer Welch. Distributed Computing. Fundamentals, Simulations,and Advanced Topics. John Wiley and Sons, Inc.

[71] Utility computing. IBM System Journal. http://www.research.ibm.com/journal/sj43-1.html

[72] IBM. SOA 和 Web Services 入门. IBM developerWorks中国.http://www.ibm.com/developerworks/cn/webservices/newto/index.html

[73] Chang Hua Sun, et al. Simplying Service Deployment with Virtual Appliances. in Proceeding of International Conference on Service Computing, SCC' 2008. July 2008: 265 272.

[74] Salesforce. 哈根达斯使用Salesforce.com构建CRM系统的成功案例. http://www.salesforce.com/customers/distribution-retail/haagen-dazs.jsp

[75] Amazon. 华盛顿邮报使用Amazon EC2进行大规模档案转换的成功案例. http://aws.amazon.com/solutions/case-studies/washington-post/

[76] GigaOM's Refresh the net Report. Why the Internet need a makeover. http://www.scribd.com/doc/3569671/GigaOMs-Refresh-the-Net-Report

[77] Cloud computing Economies of scale. http://mvdirona.com/jrh/TalksAndPapers/JamesHamilton_SMDB2009.pdf.

[78] Cloud Computing Types. Public Cloud, Hybrid Cloud, Private Cloud. http://www.circleid.com/posts/20090306_cloud_computing_types_public_hybrid_private/

[79] PARKHILL, D. The Challenge of the Computer Utility. Addison-Wesley Educational Publishers Inc., US, 1966.

[80] Wikipedia. Software Architecture. http://en.wikipedia.org/wiki/Software_architecture

[81] Edsger Dijkstra. Go-to statement considered harmful, in Commun. ACM 11 (1968), 3: 147 148.

[82] Ali Arsanjani, et al. Design an SOA solution using a reference architecture. http://www.ibm.com/developerworks/library/ar-archtemp/

[83] BBC technology analysis report. "YouTube hits 100m videos per day". http://news.bbc.co.uk/1/hi/technology/5186618.stm

[84] IBM Cloud Computing. http://www.ibm.com/cloud/

[85] IBM SAN Volume Controller. http://www-03.ibm.com/systems/storage/software/virtualization/svc/index.html

[86] Sanjay Ghemawat, Howard Gobioff and Shun-Tak Leung. The Google File System. in 19th ACM Symposium on Operating Systems Principles (SOSP 2003), Lake George, NY, October, 2003.

[87] Hadoop Distributed File System. http://hadoop.apache.org/core/docs/current/hdfs_design.html

[88] VMware Virtual Machine File System. http://www.vmware.com/products/vi/esx/vmfs.html

[89] Neiger, Gil; A. Santoni, F. Leung, D. Rodgers, R. Uhlig. Intel Virtualization Techn-ology: Hardware Support for Efficient Processor Virtualization, Intel Technology Jou-rnal(Intel)10(3), July 2008: 167-178.

[90] AMD Virtualization Technology. http://www.amd.com/us/products/technologies/virtualization/Pages/amd-v.aspx

[91] Amazon CloudWatch Monitoring Model. http://docs.amazonwebservices.com/AmazonCloudWatch/latest/DeveloperGuide/

[92] Salesforce.com Acknowledges Data Loss. http://voices.

washingtonpost.com/securityfix/2007/11/salesforcecom_acknowledges_dat.html

[93] Lei Shi, et al. Iceberg: An Image Streamer for Space and Time Efficient Provisioningof Virtual Machines. International Conference on Parallel Processing Workshops,2008: 31 38.

[94] Brendan Cully, Geoffrey Lefebvre, Dutch Meyer, Mike Feeley, Norm Hutchinson, andAndrew Warfield (April 2008). Remus: high availability via asynchronous virtual machine replication. In Proceedings of the 5th USENIX Symposium on Networked SystemsDesign and Implementation. San Francisco, California. 161 174.

[95] JDBC 4.0 Specification. http://jcp.org/en/jsr/detail?id=221

[96] Ewan Mellor, et al. Xen Management API version 1.0.6

[97] IDC Finds Cloud Computing Entering Period of Accelerating Adoption and Poised to Capture IT Spending Growth Over the Next Five Years. http://www.idc.com/getdoc.jsp?containerId= prUS21480708

[98] Google App Engine. http://code.google.com/appengine/

[99] Java Enterprise Edition (JavaEE) Specification.http://jcp.org/aboutJava/communityprocess/final/jsr244/index.html

[100] Yefim V. Natis. Reference Architecture for Multitenancy. Enterprise Computing “in the Cloud”. Gartner.com publication. ID Number G00163395

[101] Robert P. Desisto, Ben Pring. Essential SaaS Overview and 2009 Guide to SaaS Research. Gartner Research. ID Number: G00167279, 23 April 2009.

[102] Google Web Toolkit. http://code.google.com/webtoolkit/gettingstarted.html

[103] Samantha. SaaS：Gmail证明其优势 云计算迎来其辉煌. http://news.iresearch.cn/0468/20080522/81000.shtml

[104] Yefim V. Natis. Reference Architecture for Multitenancy: Enterprise Computing “in the Cloud”. Gartner Research Publication. ID number: G00163395

[105] Stefan Ried. Forrester’s SaaS Maturity Model, Transforming Vendor Strategy While Managing Customer Expectations. Forrester.com,

April 14, 2008.

[106] Lars-Olof Burchard, Matthias Hovestadt, et al. The Virtual Resource Manager: An Ar-chitecture for SLA-aware Resource Management. IEEE International Symposium on Cluster Computing and the Grid 2004, April 2004: Pages 126 133.

[107] Karl Czajkowski, Ian Foster, et al. SNAP: A Protocol for Negotiating Service LevelAgreements and Coordinating Resource Management in Distributed Systems. In 8th Workshop on Job Scheduling Strategies for Parallel Processing, Springer Verlag Berlin Heidelberg 2002: 153-183.

[108] Douglas Thain, Todd Tannenbaum, Miron Livny. Distributed Computing in Practice: the Condor experience. Concurrency and Computation: Practice and Experience.
2005, Vol.17 (No.2-4): 323 356.

[109] Armando Fox, Steven D. Gribble, Yatin Chawathe, Eric A. Brewer, and Paul Gauth-ier. Cluster-based scalable network services. In Proceedings of the 16th ACM Symp-osium on Operating System Principles, Saint-Malo, France, 1997: 7-91.

[110] Luiz A. Barroso, Jeffrey Dean, and Urs H ¨ olzle. Web search for a planet: The Google cluster architecture. IEEE Micro, April 2003, 23(2):22-28.

[111] Remzi H. Arpaci-Dusseau, Eric Anderson, Noah Treuhaft, David E. Culler, Joseph M.Hellerstein, David Patterson, and Kathy Yelick. Cluster I/O with River: Making the fast case common. In Proceedings of the Sixth Workshop on Input/Output in P-arallel and Distributed Systems (IOPADS '99), Atlanta, Georgia, May 1999: 10 22.

[112] Jeffrey Dean, Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large\Clusters. In Proceedings of the 6th conference on Symposium on Operating System Design & Implementation (2004), OSDI' 04: 137 150.

[113] Apache Hadoop. http://hadoop.apache.org/

[114] T. Johnson. Designing a distributed queue. In Proceeding of the 7th IEEE Symposi-um on Parallel and Distributed Processing, 304 311.

[115] Java Message Service (JMS) Specification. http://jcp.org/en/jsr/detail?id=914

[116] Apache ActiveMQ. http://activemq.apache.org/

[117] IBM Websphere MQ. http://www-01.ibm.com/software/integration/wmq/

[118] Ellard T.Roush, Roy H. Compbel. Fast Dynamic Process Migration. In proceedings of the 6th IEEE International Conference on Distributed Computing System, 1996: 637-645.

[119] Brendan Cully, Geoffrey Lefebvre, et al. Remus: High Availability via AsynchronousVirtual Machine Replication. In Proceedings of the 5th USENIX Symposium on Networked System Design and Implementation NSDI' 08, 2008: 161-174.

[120] Chandramohan A. Thekkath, Timothy Mann and Edward K. Lee. Frangipani: A Sca-lable Distributed File System, In Proceedings of the 16th ACM Symposium on Operating Systems Principles, SOSP' 97, 1997: 224-237.

[121] Björn Grönvall, Assar Westerlund, and Stephen Pink. The Design of a Multicast-based Distributed File System. In Proceedings of the 3rd Symposium on Operating Systems Design and Implementation (OSDI' 96): 251-264.

[122] Sage A. Weil, Scott A. Brandt, Ethan L. Miller, Darrell D.E. Long and Carlos Mal-tzahn. Ceph: A Scalable, High-performance Distributed File System. In Proceedings of the 7th Symposium on Operating Systems Design and Implementation OSDI' 06, 2006: 307-320.

[123] S Ghemawat, H Gobioff, S. T. Leung. The Google File System. In Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP 2003), Lake George, NY, October 2003: 29-43.

[124] F Chang, J Dean, S Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Cha-ndra, A. Fikes, R. E. Gruber. Bigtable: A Distributed Storage System for StructuredData. OSDI 2006.

[125] G DeCandia, D Hastorun, M Jampani, G Kakulapati, A Lakshman, A Pilchin, S Si-vasubramanian, P Vosshall, W Vogels. Dynamo: Amazon's Highly Available Key-value Store. SOSP 2007.

[126] H Yang, A Dasdan, RL Hsiao, DS Parker. Map-Reduce-Merge: Simplified RelationalData Processing on Large Clusters. Proceedings of the 2007 ACM SIGMOD.

[127] Yu Hui Wu, Zhi Le Zou, et al. SPA: A Comprehensive Framework for Hybrid Sol-ution Provisioning. In Proceedings of the IEEE 7th International Conference on WebSe-rvices, ICWS 2009, July 6-10, 2009.

[128] IBM. 全新企业级数据中心发展之路 - IT优化. http://www-900.ibm.com/cn/support/guide/itoebook7_1.shtml

[129] Cloudbook: RC2. http://www.cloudbook.net/private-cloud/ibm

[130] IBM News. IBM 服务管理决胜企业动态架构. http://www-01.ibm.com/software/swnews/swnews.nsf/n/ydlz7sg9er?OpenDocument&Site=default

[131] Salesforce: Force.com Platform. http://www.salesforce.com/platform/what-is-it.jsp

[132] Force.com. Developer Core Resource Library. http://wiki.developerforce.com/index.php/DeveloperCoreResources

[133] Force.com. An Overview of Force.com Security. http://wiki.developerforce.com/index.php/An_Overview_of_Force.com_Security

[134] Force.com. Database Services. http://wiki.developerforce.com/index.php/Database_Services

[135] Force.com. Create and Run any Application, On Demand. http://wiki.developerforce.com/index.php/Force.com:_Create_and_Run_any_Application,_On_Demand

[136] Google App Engine Blog. Back to the Future for Data Storage. http://googleappengine.blogspot.com/2009/02/back-to-future-for-data-storage.html

[137] Wikipedia. Optimistic concurrency control. http://en.wikipedia.org/wiki/Optimistic_concurrency_control.....

[138] Google Code. Queries and Indexes. http://code.google.com/intl/en/appengine/docs/python/datastore/queriesandindexes.html

[139] Google Code. Datastore Python API Overview. http://code.google.

com/intl/en/appengine/docs/python/datastore/overview.html

[140] Google Code. The Administration Console. http://code.google.com/intl/en/appengine/docs/theadminconsole.html

[141] Online collaboration. LotusLive. https://www.lotuslive.com/services

[142] Microsoft's Azure cloud platform: A guide for the perplexed. http://blogs.zdnet.com/microsoft/?p=1671

[143] Microsoft. Windows Azure Platform. http://www.microsoft.com/azure/default.mspx

[144] David Chappell. Introducing the Azure Service Platform, May, 2009. http://www.microsoft.com/presspass/events/pdc/docs/AzureServicesPlatform.pdf

[145] David Chappell. Introducing Windows Azure. http://download.microsoft.com/download/0/C/0/0C051A30-F863-47DF-BC53-9C3CFA88E3CA/Windows%20Azure%20David%20Chappell%20White%20Paper%20March%2009.pdf

[146] Wikipedia. Amazon.com. http://en.wikipedia.org/wiki/Amazon.com

[147] Amazon web services. Amazon Simple Storage Service (Amazon S3). http://aws.amazon.com/s3/

[148] Amazon Web Services. Amazon Simple Storate Service Getting Started Guide(API Version 2006-03-01). http://docs.amazonwebservices.com/AmazonS3/latest/gsg/

[149] James Murty. Programming Amazon Web Services. O'REILY, 2008.

[150] Amazon Web Services. Amazon SimpleDB. http://aws.amazon.com/simpledb/

[151] Amazon Web Services. Amazon SimpleDB Getting Started Guide (API Version 2009-04-15), http://docs.amazonwebservices.com/AmazonSimpleDB/latest/GettingStartedGuide/

[152] Amazon Web Services. Amazon Simple Queue Service (Amazon SQS). http://aws.amazon.com/sqs/

[153] Amazon Web Services. Amazon Simple Queue Service Getting Started Guide (API Version 2009-02-01). http://docs.amazonwebservices.com/AWSSimpleQueueService/latest/SQSGettingStartedGuide/

[154] Amazon Web Services. Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2/

[155] Amazon Web Services. Amazon Elastic Compute Cloud Getting Started Guide (APIVersion 2009-04-04). http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/

[156] Jinesh Varia. Cloud Architectures. Amazon Web Services white paper. http://jineshvaria.s3.amazonaws.com/public/cloudarchitectures-varia.pdf

[157] Thomas L. Friedman, The World Is Flat: A Brief History of the Twenty-first Century, April, 2005.

[158] Jin Xing and etc., Reinventing virtual appliances, Journal of Research and Development, 2009.

[159] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris,A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen andthe Art of Virtualization. In Proceedings of the 19th ACMSOSP, pages 164 177, October 2003.

[160] B. Clark, T. Deshane, E. Dow, S Evanchik, M. Finlayson,J. Herne, and J.N. Matthews. Xen and the art of repeated re-search. In Proceedings of the Usenix annual technical con-ference, Freenix track, July 2004.

[161] D F. Parkhill, The challenge of the computer utility, Addison-Wesley, 1966.

# 索　引

Broadview®
www.broadview.com.cn
博文视点·IT出版旗舰品牌
技术凝聚实力·专业创新出版

## 本书将为您精彩阐释：

- 未来的数据中心是什么样子？
- 虚拟化是什么，它究竟能带来哪些好处？
- 业界有哪些虚拟化厂商，他们的技术区别是什么？
- 到底什么是云计算，它会使信息产业发生革命性的改变吗？
- 云计算如何为各种企业和个人用户创造价值？
- 虚拟化技术和云计算有什么关系？
- 如何在数据中心构建云计算环境，需要哪些关键技术？
- 云计算会给信息产业带来哪些机遇和挑战？
- 业界有哪些云计算厂商，他们都有什么产品和解决方案？

上架建议：计算机>云计算

ISBN 978-7-121-09678-5

9 787121 096785 >

定价：45.00元

策划编辑：郭　立　刘　皎
责任编辑：郭　立
文字编辑：刘　皎
责任美编：李　玲