

探究核心技术 ▪ 总结实践方法 ▪ 掌舵智慧引擎 ▪ 制胜行业转型

云计算宝典

技术与实践

『虚拟化与云计算』小组 著

作者团队再次倾情奉献，全面整合畅销书《虚拟化与云计算》和《云计算实践之道》的精华内容，并增添、重写、更新重要资讯，以满足读者不同层次的期望。

综合展示全球领先IT企业的云计算行业实践和国内一线互联网公司的技术视角。

冀望各位读者都能从中获益，在IT转型的重要时刻，抓住时代机遇，实现人生的飞跃。

寄语



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

您可按需选读，亦可全书通读：



所获奖项：

- 「虚拟化与云计算」小组荣获2009年电子工业出版社优秀作者奖
- 《虚拟化与云计算》荣获51CTO 2009年最受读者喜爱的IT原创图书奖
- 作者团队的《云端策略》（《虚拟化与云计算》之繁体中文修订版）荣获台湾地区2010年经济部选定金书奖

探究核心技术 ■ 总结实践方法 ■ 掌舵智慧引擎 ■ 制胜行业转型

上架建议： 云计算



策划编辑：刘 皎
责任编辑：高洪霞
封面设计：侯士卿

本书贴有激光防伪标志，凡没有防伪标志者，属盗版图书。



ISBN 978-7-121-14103-4



9 787121 141034 >

定价：75.00元

云计算宝典

技术与实践

主审：陈 滢

著者：「虚拟化与云计算」小组

王庆波 金 滓 何 乐 赵 阳

邹志乐 吴玉会 杨 林 田瑞雄

谢苏苏 操保华 王 仕

(以加入研究团队的时间先后为序)



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

探究核心技术，总结实践方法。本书四篇十二章，对云计算进行了系统的探讨。战略蓝图篇以概述云计算为起点，纵览其在一系列重要行业中的应用场景和实践方针，继而深入实施步骤，为云计算产业参与者勾画了蓝图。技术基石篇首先描述数据中心的核心功能，管理维护方法，以及新一代数据中心的需求和挑战，随后介绍云计算的技术基础：服务器虚拟化和虚拟器件。系统架构篇从横向和纵向两个维度定义了云架构，在基础设施即服务、平台即服务和软件即服务的层面进行了深入的技术探讨，并给出相关的参考实现。业界动态篇勾勒了生态系统的动态和趋势，从虚拟化和云计算两个维度介绍了业界的基本情况和主流产品。

通过阅读本书，您可以系统地了解云计算的产生背景、发展现状、技术要点和未来趋势，更加准确地把握业界前沿的科技和理念，认清信息技术发展的大脉络，形成适应于产业未来的大局观。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，储备适用于下一代信息产业的技能和智慧。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

云计算宝典：技术与实践/虚拟化与云计算小组著. —北京：电子工业出版社，2011.9

ISBN 978-7-121-14103-4

I. ①云… II. ①虚… III. ①计算机网络 IV. ①TP393

中国版本图书馆CIP数据核字(2011)第138980号

策划编辑：刘 皎

责任编辑：高洪霞

印 刷：北京东光印刷厂

装 订：三河市皇庄路通装订厂

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：787×980 1/16 印张：27.75 字数：677千字

印 次：2011年9月第1次印刷

印 数：4000册 定价：75.00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。

作者序

本书作者大多曾供职于IBM中国研究院虚拟化与云计算研究部。自2006年成立以来，该部门一直致力于研究如何使用虚拟化和云计算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。我们开创了应用虚拟器件技术及云平台技术来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和产业界的标准。我们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云平台解决方案，并通过快捷部署激活工具简化应用上线的过程，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。

作为长期工作在产业前沿的研究团队，让更多的技术人员深刻理解虚拟化以及云计算技术是我们肩负的一份责任。我们到国内各大高校做了多场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。于是，我们在2009年10月写作完成了小组的第一本专著《虚拟化与云计算》，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。该书获得了51CTO 2009年最受读者喜爱的IT原创图书奖，我们也获得了电子工业出版社2009年优秀作者奖。2010年，我们在台湾地区出版了《虚拟化与云计算》的繁体修订版《雲端策略》，之后又创作完成小组的第二本专著《云计算实践之道——战略蓝图与技术架构》，书中我们加深阐述了对于云计算本质及优势的理解，增加了云计算的行业实践以及云计算的实施两大部分内容，从而解答读者对于云计算的收益、实施场景、实施方法以及实施要点等问题。

自这三本书出版以来，业界对云计算的认识和实践已经出现了很大变化。一方面，

目 录

CONTENTS

第一篇 战略蓝图篇

第1章 云计算概论	003
1.1 云计算的概念.....	003
1.1.1 走近云计算.....	004
1.1.2 云计算的定义.....	007
1.1.3 云计算的特征.....	010
1.1.4 云计算的分类.....	012
1.1.5 相关概念辨析.....	017
1.2 云计算的优势.....	020
1.2.1 优化产业布局.....	020
1.2.2 推进专业分工.....	021
1.2.3 提升资源利用率.....	022
1.2.4 减少初期投资.....	024
1.2.5 降低运营成本.....	026
1.2.6 产生新创价值.....	026
1.3 云计算产生的原动力.....	027
1.3.1 芯片与硬件技术.....	029
1.3.2 资源虚拟化.....	029
1.3.3 面向服务架构.....	030
1.3.4 软件即服务.....	031
1.3.5 互联网技术.....	031
1.3.6 Web 2.0技术.....	032
1.4 云计算带来的变革.....	033
1.4.1 大势所趋的转型.....	033
1.4.2 新兴的产业链.....	034
1.5 小结.....	039
第2章 云计算的行业实践	040
2.1 概述.....	041

2.2	云计算在公共服务行业的应用	043
2.3	云计算在银行业的应用	047
2.3.1	中间业务创新	047
2.3.2	核心业务创新	049
2.3.3	开发测试业务创新	051
2.4	云计算在电信行业的应用	052
2.5	云计算在物流行业的应用	055
2.6	云计算在医疗行业的应用	057
2.7	云计算在制造行业的应用	059
2.8	云计算在教育科研领域的应用	063
2.8.1	云计算在课堂教学领域的应用	063
2.8.2	云计算在教学实验中的应用	064
2.8.3	云计算在教辅领域的应用	065
2.8.4	云计算在促进科研合作中的应用	065
2.9	小结	067

第3章 云计算的实施 070

3.1	云计算的实施要点	071
3.2	企业实施云计算的战略规划	072
3.2.1	战略规划概述	073
3.2.2	价值分析	074
3.2.3	风险评估	075
3.2.4	战略定位分析	077
3.2.5	业务适应性分析	078
3.3	企业云计算业务的实施	084
3.3.1	实施基础设施层云计算业务	085
3.3.2	实施平台层云计算业务	086
3.3.3	实施应用层云计算业务	090
3.4	云计算提供商的业务模型	090
3.4.1	业务模型设计	091
3.4.2	业务模型示例	094

3.5 云计算提供商的平台构建.....	096
3.6 云计算平台的运维管理.....	098
3.6.1 运维管理的目标.....	099
3.6.2 运维管理的核心.....	100
3.6.3 运维管理的平台.....	102
3.6.4 平台信息安全管理.....	105
3.7 小结.....	111

第二篇 技术基石篇

第4章 新一代绿色数据中心..... 113

4.1 数据中心概述.....	114
4.1.1 数据中心的概念.....	114
4.1.2 数据中心的发展过程.....	115
4.1.3 数据中心的分类与分级.....	116
4.2 数据中心的设计和构建.....	117
4.2.1 总体设计.....	117
4.2.2 建筑的设计与构建.....	118
4.2.3 基础设施的设计与构建.....	120
4.2.4 数据中心上线.....	122
4.3 数据中心的管理和维护.....	125
4.3.1 硬件的管理和维护.....	126
4.3.2 软件的管理和维护.....	126
4.3.3 数据的管理和维护.....	127
4.3.4 资源管理.....	128
4.3.5 安全管理.....	129
4.4 新一代数据中心的需求.....	130
4.4.1 合理规划.....	130
4.4.2 流程化.....	131
4.4.3 可管理性.....	133
4.4.4 可伸缩性.....	134
4.4.5 可靠性.....	135

4.5	绿色数据中心.....	136
4.5.1	经济型数据中心.....	136
4.5.2	数据中心能效分析.....	137
4.6	小结.....	140
第 5 章 虚拟化概论.....		142
5.1	虚拟化的定义.....	143
5.1.1	走近虚拟化.....	143
5.1.2	虚拟化的定义.....	144
5.1.3	虚拟化的常见类型.....	146
5.2	服务器虚拟化.....	149
5.2.1	基本概念.....	149
5.2.2	典型实现.....	151
5.2.3	关键特性.....	152
5.2.4	核心技术.....	153
5.2.5	性能分析.....	161
5.2.6	技术优势.....	164
5.3	其他虚拟化技术.....	167
5.3.1	网络虚拟化.....	167
5.3.2	存储虚拟化.....	168
5.3.3	桌面虚拟化.....	169
5.3.4	应用虚拟化.....	170
5.4	小结.....	172
第 6 章 虚拟化管理.....		173
6.1	创建虚拟化解决方案.....	174
6.1.1	创建基本虚拟镜像.....	174
6.1.2	创建虚拟器件镜像.....	176
6.1.3	发布虚拟器件镜像.....	180
6.1.4	管理虚拟器件镜像.....	182

6.1.5	迁移到虚拟化环境.....	183
6.2	部署虚拟化解决方案.....	185
6.2.1	规划部署环境.....	185
6.2.2	部署虚拟器件.....	187
6.2.3	激活虚拟器件.....	191
6.3	管理虚拟化解决方案.....	192
6.3.1	集中监控.....	193
6.3.2	快捷管理.....	194
6.3.3	动态优化.....	196
6.3.4	高效备份.....	198
6.4	小结.....	200

第三篇 系统架构篇

第7章 云架构..... 202

7.1	云架构的层次.....	203
7.1.1	云架构的基本层次.....	203
7.1.2	云架构的服务层次.....	205
7.2	云架构的特性.....	207
7.2.1	大规模.....	207
7.2.2	高可用.....	209
7.2.3	可伸缩.....	210
7.2.4	高性能.....	210
7.3	云架构的准则.....	211
7.3.1	信息安全与保密.....	211
7.3.2	许可证与计费.....	213
7.3.3	集成与标准化.....	215
7.4	小结.....	217

第8章 基础设施即服务..... 218

8.1	概述.....	219
8.2	服务模型与接口.....	219

8.3	计算即服务	222
8.3.1	服务模型及接口	222
8.3.2	关键技术	224
8.3.3	参考实现	229
8.4	存储即服务	238
8.4.1	服务模型及接口	239
8.4.2	关键技术	240
8.4.3	参考实现	245
8.5	网络即服务	247
8.5.1	服务模型及接口	247
8.5.2	关键技术	249
8.6	小结	253

第9章 平台即服务 254

9.1	概述	255
9.1.1	驱动力	255
9.1.2	主流类型	257
9.1.3	功能角色	258
9.2	核心系统	260
9.2.1	简化的应用开发和部署模型	260
9.2.2	自动的资源获取和应用激活	262
9.2.3	自动的应用运行管理	263
9.2.4	平台级优化	265
9.3	扩展系统	266
9.3.1	非关系型数据存取	266
9.3.2	大规模消息通信	272
9.3.3	海量数据分析	274
9.4	参考实现	278
9.4.1	事务处理类	279
9.4.2	数据分析类	283

9.5 小结.....	291
-------------	-----

第 10 章 软件即服务..... 292

10.1 概述.....	293
10.1.1 生态系统与实现层次.....	294
10.1.2 技术发展历程.....	295
10.2 支撑平台.....	296
10.2.1 支撑平台的架构.....	296
10.2.2 支撑平台的关键技术.....	298
10.2.3 支撑平台的参考实现.....	309
10.3 云应用.....	311
10.3.1 云应用的特征.....	311
10.3.2 云应用的分类.....	312
10.3.3 云应用的典型示例.....	314
10.4 小结.....	318

第四篇 业界动态篇

第 11 章 虚拟化的业界动态..... 320

11.1 IBM.....	321
11.1.1 概述.....	321
11.1.2 z系列服务器.....	323
11.1.3 p系列服务器.....	325
11.1.4 虚拟化管理.....	328
11.2 VMware.....	330
11.2.1 概述.....	330
11.2.2 数据中心虚拟化.....	331
11.2.3 桌面和应用虚拟化.....	334
11.2.4 虚拟化辅助工具.....	335
11.3 Xen/Citrix.....	336

11.3.1	概述	336
11.3.2	服务器虚拟化	337
11.3.3	应用虚拟化	338
11.3.4	桌面虚拟化	338
11.4	KVM/Red Hat	339
11.4.1	概述	339
11.4.2	架构	340
11.4.3	展望	342
11.5	Microsoft	342
11.5.1	概述	342
11.5.2	服务器虚拟化	344
11.5.3	应用虚拟化	344
11.5.4	桌面虚拟化	345
11.5.5	虚拟化管理	346
11.6	小结	346

第 12 章 云计算的业界动态 348

12.1	IBM	349
12.1.1	概述	349
12.1.2	IBM 云业务咨询服务	351
12.1.3	IBM 云计算基础架构策略和计划服务	353
12.1.4	IBM Smart Business Development and Test Cloud	355
12.1.5	IBM Smart Business Desktop Cloud	358
12.1.6	IBM Smart Business Analytics Cloud	362
12.1.7	IBM Smart Business Storage Cloud	363
12.1.8	IBM LotusLive	365
12.1.9	IBM TSAM	366
12.1.10	IBM WebSphere CloudBurst Appliance	369
12.1.11	IBM System Director	372
12.2	Amazon	376
12.2.1	概述	376
12.2.2	Amazon S3	378
12.2.3	Amazon SimpleDB	379

12.2.4	Amazon RDS.....	380
12.2.5	Amazon SQS.....	381
12.2.6	Amazon EC2.....	382
12.3	Google.....	384
12.3.1	概述.....	384
12.3.2	分布式存储服务.....	385
12.3.3	应用程序运行时环境.....	386
12.3.4	应用开发套件.....	386
12.3.5	云应用.....	387
12.4	Salesforce.com.....	388
12.4.1	概述.....	388
12.4.2	基础服务.....	388
12.4.3	数据库服务.....	389
12.4.4	应用开发服务.....	390
12.4.5	应用打包服务.....	391
12.5	Microsoft.....	392
12.5.1	Windows Azure platform.....	393
12.5.2	Live服务.....	398
12.6	开源云计算.....	400
12.6.1	Eucalyptus.....	400
12.6.2	NoSQL.....	404
12.6.3	OpenStack.....	407
12.6.4	Nimbus.....	408
12.6.5	AppScale.....	409
12.7	小结.....	411
	附录A: 参考文献.....	413



第一篇 战略蓝图篇

第1章 云计算概论

第2章 云计算的行业实践

第3章 云计算的实施

「第一篇」
战略蓝图篇



第 1 章

云计算概论

- 1.1 云计算的概念
- 1.2 云计算的优势
- 1.3 云计算产生的原动力
- 1.4 云计算带来的变革
- 1.5 小结

浙
大
知
行

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？各种新的IT技术对各行业将会产生怎样的影响？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，势不可当。本章将解释云计算的确切含义与分类，分析云计算的优势及其带来的变革，并阐述云计算的来龙去脉。

1.1 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对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。

1.1.1 走近云计算

1.1.1.1 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个虚拟服务器案例，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9所研究院，汇聚了3000多位顶尖的科学家和研究员。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。在2009年，共有4914项美国专利在IBM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全球各地，处于不同的时区，给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。

1.1.1.2 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。

此外，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业务本身，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，可能会多次出现需求→设计→实施→需求变更→再设计→再实施的循环。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。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9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。

1.1.2 云计算的定义

1.1.2.1 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时就选择了“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1.1左侧所示。



图1.1 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能力。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1.1右侧我们可以看出，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。

1.1.2.2 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，我们先列举一些被人们普遍认可的云计算定义，然后再给出本书的定义。

在维基百科（Wikipedia.org）中，截止到2010年7月，“云计算”的词条被表述为是一

种基于互联网的计算，在其中共享的资源、软件和信息以一种按需的方式提供给计算机和设备，就如同日常生活中的电网一样。

专业的IT名词百科Whatis.com援引来自SearchCloudComputing.com的定义，广义地将云计算解释为一切能够通过互联网提供的服务，这些服务被划分为三个层次：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）。

在美国加州大学伯克利分校（UC Berkeley）发表的一篇关于云计算的报告中，云计算既指在互联网上以服务形式提供的应用，也指在数据中心里提供这些服务的硬件和软件，而这些数据中心的硬件和软件则被称为云。

《商业周刊》（BusinessWeek.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互联网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的，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认为云计算是一种革新的信息技术与商业服务的消费与交付模式。在这种模式中，用户可以采用按需的自助模式，通过访问无处不在的网络，获得来自于与地理无关的资源池中被快速分配的资源，并按实际使用情况付费。本书沿用IBM的定义。这种模式的主体是所有连接着互联网的实体，可以是人、设备或程序。这种模式的客体是服务本身，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与商业服务。这种模式的核心原则是：

硬件和软件都是资源并被封装为服务，用户可以通过网络按需地访问和使用。

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。

1.1.3 云计算的特征

在云计算的定义中有四个关键要素，如图1.2所示。

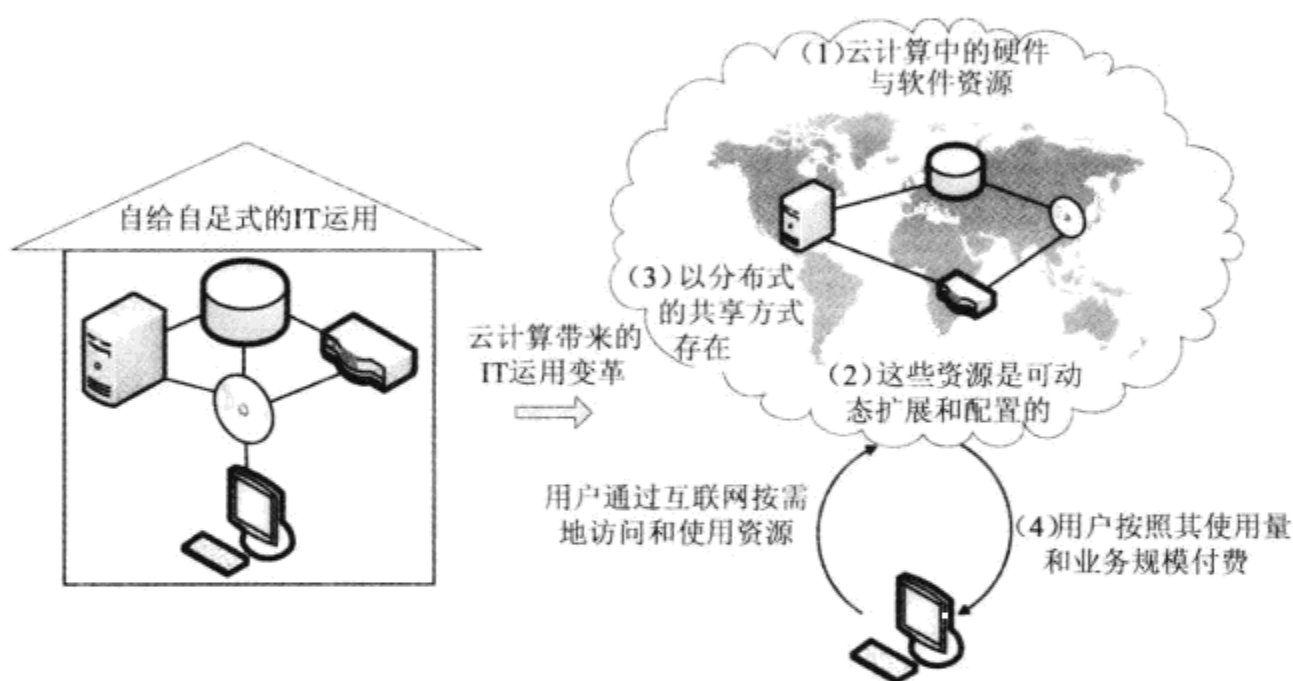


图1.2 云计算的特征

1. 硬件和软件都是资源，通过网络以服务的方式提供给用户

正如上一节所描述的，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将专业的客户关系管理应用模块打包成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。

2. 这些资源都可以根据需要进行动态扩展和配置

例如在前面的典型案例中，Amazon EC2可以在极短的时间内为华盛顿邮报初始化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。

3. 这些资源在物理上以分布式的方式存在，为云中的用户所共享，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。

在前面介绍的典型案例四中，IBM公司在世界范围内共拥有9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。

4. 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，华盛顿邮报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。虽然华盛顿邮报没有足够的运算处理能力，但是云给了它强大的资源来快速完成任务，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，华盛顿邮报在9小时内完成档案的转换工作将是不可能的。

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。

1.1.4 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，向用户提供；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，用户都获取了云中的服务，快速、高效地完成了工作；不同点是，用户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。

1.1.4.1 按服务类型分类

所谓云计算的服务类型，就是指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，云计算可以按照服务类型分为以下三类，如图1.3所示。

1. 基础设施云（Infrastructure Cloud）

例如在前面的案例一中提到的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资源、存储资源和网络资源，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，不做进一步任何应用类型的假设。



图1.3 云计算的服务类型

2. 平台云 (Platform Cloud)

在前面案例二中提到的Google App Engine就是平台云。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。

3. 应用云 (Application Cloud)

例如在前面案例三中提到的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1.1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4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1.1 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

1.1.4.2 按服务方式分类

云计算作为一种革新性的计算模式，虽然具有许多现有计算模式所不具备的优势（云计算带来的优势将在后面进行具体分析），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。其次就是可靠性问题，例如银行希望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部门完全被公司掌握，不受外界干扰和控制。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，业界以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1.4所示。用户可以根据需求选择适合自己的云计算模式。



图1.4 云计算的服务方式

1. 公有云 (Public Cloud)

公有云是由若干企业和用户共同使用的云环境，IT业务和功能以服务的方式，通过互联网来为广泛的外部用户提供；用户无须具备针对该服务在技术层面的知识，无须雇

佣相关的技术专家，无须拥有或管理所需的IT基础设施。我们前面所列举的Amazon EC2、Google AppEngine和Salesforce.com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。

2. 私有云 (Private Cloud)

私有云是由某个企业独立构建和使用的云环境，IT能力通过企业内部网，在防火墙内以服务的形式为企业内部用户提供；私有云的所有者不与其他企业或组织共享任何资源，例如我们前面提及的IBM RC2。私有云是企业或组织所专有的云计算环境。在其中，用户是这个企业或组织的内部成员，他们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。

3. 混合云 (Hybird Cloud)

混合云是整合了公有云与私有云所提供服务的云环境。用户根据自身因素和业务需求选择合适的整合方式，制订其使用混合云的规则和策略。在这里，自身因素是指用户本身所面临的限制与约束，如信息安全的要求、任务的关键程度和现有基础设施的情况等，而业务需求是指用户期望从云环境中所获得的服务类型。有研究表明，例如网络会议、帮助与培训系统这样的服务适合于从公有云中获得；例如数据仓库、分析与决策系统这样的服务适合于从私有云中获得。

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日常事务性的支撑性应用部署在公有云上，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。

值得注意的是，虽然私有云能够为企业或组织创建一个独占的云环境，具有防火墙内的信息安全保障，提供资源与服务共享的便利，但是拥有与运维一个私有云需要较高的资金投入与持续的技术支持，即便是实力雄厚的公司也会力不从心。同样，虽然公有云能够为用户快速而便捷地提供IT能力，但是有些企业和组织希望能够获得更强的私密性，因

此，在现实的生产环境中，云的私有性和公有性并不是泾渭分明的，而是存在着多种逐级过渡的方案，如图1.5所示。

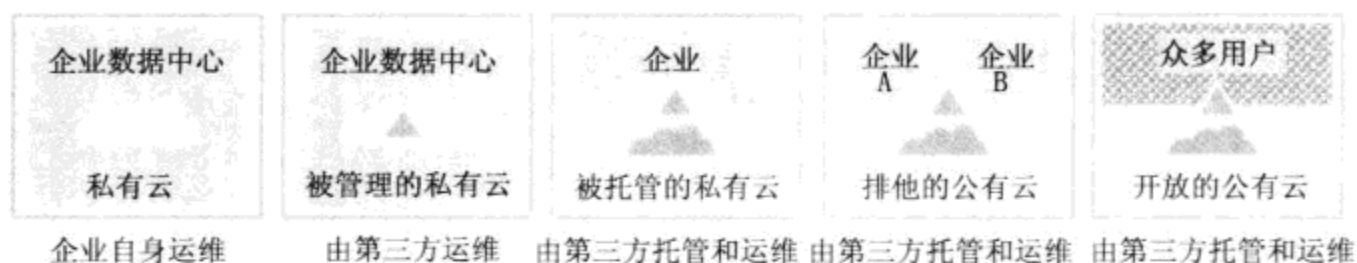


图1.5 私有云至公有云的逐级过渡

除了完全由自己拥有和运维的私有云外，用户还可以选择“被管理的私有云”和“被托管的私有云”两种提供模式。在被管理的私有云中，承载云环境的IT设备和基础设施仍由所属的企业或组织拥有，在物理上位于企业的数据中心内，但其私有云的创建和运维将由专业的第三方公司来完成。一般来说，这样的第三方公司常常会通过以下步骤来帮助客户完成私有云的搭建：第一，将客户现有的物理资源通过虚拟化技术进行逻辑化，形成便于划分的资源池；第二，在该逻辑资源池上创建业务应用，并订立服务目录以便使用者浏览；第三，为业务应用提供自助访问接口和用量计费功能，服务上线并为私有云所属的企业或组织内用户所用。此后，该第三方公司还将为客户持续地提供在运维上的支持，如安全管理、业务升级、新服务上线等。

与被管理的私有云相同，被托管的私有云的创建与运维将由第三方公司来完成。相比前者更进一步，如果客户选择后者作为自己拥有私有云的模式，它将不再需要建设自己的数据中心，云环境所需的IT设备和基础设施将被托管在由第三方公司提供的专业数据中心内，并可根据合同的订立来保证客户在该数据中心内对资源在物理上或逻辑上的独占性。这种独占性是该模式与公有云的本质区别。在公有云中，不同客户需通过多租户（Multi Tenancy）技术来共享底层资源，有关该技术细节将在第10章中介绍。

同样，用户对于公有云的选择还可以分为排他的公有云和开放的公有云两种。在排他云中，云服务的提供者和使用者的不是同一个企业，但它们事先知道谁会提供何种服务，谁会使用何种服务，它们通过线下的协商确定服务价格和服务质量。排他云通常出现在企业的联盟中，例如：某大企业与它的众多供应商和业务伙伴间可以建立排他云，大企业为供应商们提供云服务；某一行业联盟中的企业间可以建立排他云，比如：航空公司、酒店、旅行社等组成的联盟。

在开放的公有云中，服务的使用者和提供者在服务预订前彼此不知晓对方，他们的关系是通过在线服务订阅的方式确立的。服务条款通常是由服务提供方预先定义和控制的，而服务价格和服务质量约定也是自动的和标准化的，由服务提供方预先设定。

综上，从私有云到公有云，第三方公司能够为客户提供不同深度的自底向上的整合服务，帮助用户便捷可靠地获得私有云，同时有效减轻其建设数据中心、购置基础设施和运维云环境的负担。

1.1.5 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，云计算是从这些技术中逐渐演进而来的，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。

1.1.5.1 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，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集群上。云计算则更加强调用户通过互联网使用云服务，并在云中利用虚拟化进行大规模的系统资源抽象和管理。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。

1.1.5.2 网络计算

网络计算（Grid Computing）是一种分布式计算模式。网络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。对于使用网络的最终用户或应用程序来说，网络看起来就像是一个拥有超强性能的虚拟计算机。网络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，网络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络的框架来构建自己的网络系统，并对其进行管理，执行计算任务。而云计算则不同，用户只需要使用云中的资源，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。

另外由于网络因素，网络计算的终端通常只进行小量的计算和数据传输。因此网格计算通常不适合需要大规模数据传输的计算。

1.1.5.3 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）一样地供应和收费。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，可行性更强。

1.1.5.4 物联网

物联网（Internet of Things, IoT）是一个将人、物理实体和信息系统互联起来的遍布全球的系统。它通过可扩展、价格可以接受的技术，如大范围的数据搜集、智能网络、预测分析和深度优化等，来更好地管理物理世界。物联网是当前重要的创新领域。国务院总理温家宝2009年11月3日在人民大会堂向首都科技界发表题为《让科技引领中国可持续发展》的讲话中明确指出：信息网络产业是世界经济复苏的重要驱动力。全球互联网正在向下一代升级，传感网和物联网方兴未艾。温总理的讲话将物联网提升到国家战略的高度，物联网的应用也日趋丰富，如高速公路不停车收费系统、公路铁路车辆调度系统、物流货品追踪管理、学生进出校门管理、手机移动支付系统等，都是基于物联网理论的应用探索。

物联网的核心和基础在于物品与物品之间的互联。相对于传统的互联网，物联网将计算机间的互联互通延伸和扩展到物与物之间。云计算与物联网在概念上有很强的关联性。我们可以将物联网看做是处于前端的传感器与网络设备、处于核心的云计算海量数据处理平台和处于上层的应用系统这三者的结合体。

云计算作为物联网数据处理的核心平台，适于处理物联网中地域分散、数据海量、动态性和虚拟性强的应用场景。它能够促进物联网底层传感数据的共享，为分析与优化提供

超级计算能力，从而更高效地提供更可靠的服务。如果将物联网比喻为人体，那么传感器就如同感知器官，网络就如同神经系统，云计算就如同大脑。传感器所获得的物理世界的信息通过网络汇聚到云中，通过云计算提供的处理、存储和共享能力，进行有针对性的调优，再通过一定的反馈机制作用于物理世界，使其更加智慧而有效地运行。可见，物联网与云计算是相辅相成的概念。云计算为物联网提供了使其发挥效用的核心能力，物联网为云计算提供了宽广而前景光明的舞台。

1.2 云计算的优势

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，必然有其技术的进步和独到的优势，才能在IT这个高速发展的产业里成长起来。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本质优势。相信读者阅读过本节后，将对云计算的未来充满信心。

1.2.1 优化产业布局

正如前面所述，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40余座数据中心，在这些数

据中心里同时运行着数百万台服务器。据估计，仅以该公司在美国华盛顿州Dalles建设的数据中心为例，其电力消耗就高达约103兆瓦，相当于82000户家庭一年的用电量。

电力作为一种传统资源，分布很不均匀。由于自然条件和政策法规的影响，各地的电价具有很大差异，如表1.2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。

表1.2 美国部分地区工业电价比较（2008年）

电价（美分/千瓦时）	地点	可能的原因
4.48	爱达荷州	丰富的水电资源
10.04	加州	电网远距离传输，州法律禁止火电厂的建造
26.05	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产业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。

1.2.2 推进专业分工

正如1.2.1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，提供单位计算力所需的成本更低廉，如表1.3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1.3 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

注：PUE即Power Usage Effectiveness，是数据中心总设备能耗与IT设备能耗的比值，PUE越接近1表明数据中心的能效水平越好。

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。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。

1.2.3 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。

如图1.6所示是一个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。

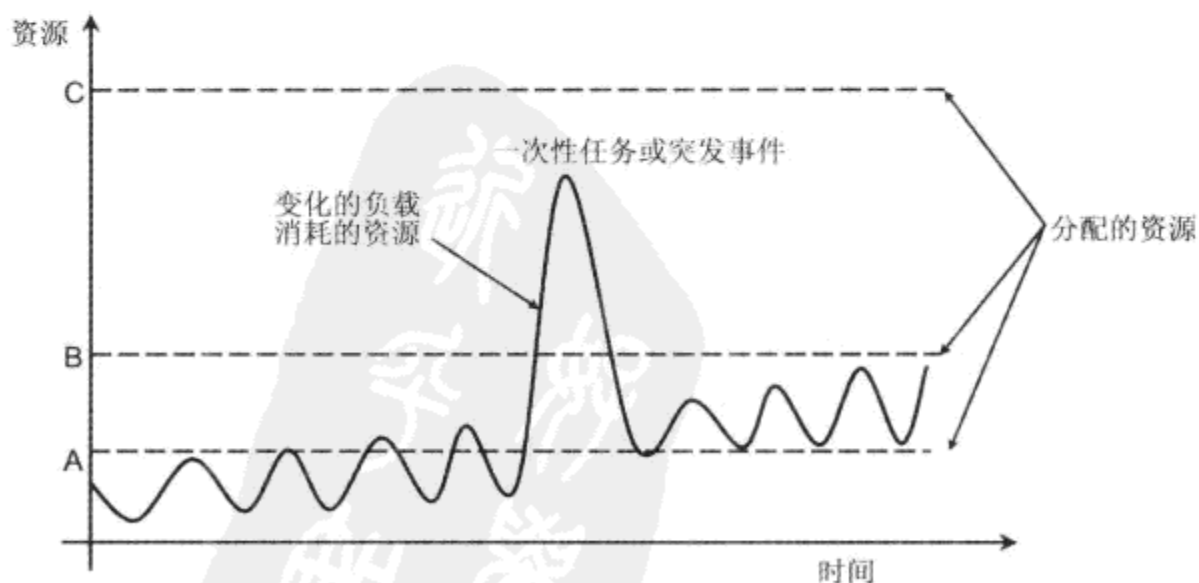


图1.6 典型的业务系统负载变化及传统的资源分配方式

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。

1.2.4 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。企业无须拥有硬件，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需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：

(1) 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；

(2) 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1.4所示列出了Amazon公司的AWS业务为美国东海岸用户提供的打包计算资源实例及计价标准。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，并根据占用资源的情况分为若干等级，各个等级的计价有所不同。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。

表1.4 Amazon公司提供的配置实例和收费标准（美国东海岸，2010年7月）

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	0.085美元/小时	0.12美元/小时
	大	7.5GB内存，4个EC2计算单元，850GB存储，64位平台	0.34美元/小时	0.48美元/小时
	超大	15GB内存，8个EC2计算单元，1690GB存储，64位平台	0.68美元/小时	0.96美元/小时
高CPU	中	1.7GB内存，5个EC2计算单元，350GB存储，32位平台	0.17美元/小时	0.29美元/小时
	超大	7GB内存，20个EC2计算单元，1690GB存储，64位平台	0.68美元/小时	1.16美元/小时
高内存	超大	17.1GB内存，6.5个EC2计算单元，420GB存储，64位平台	0.5美元/小时	0.62美元/小时
	双倍超大	34.2GB内存，13个EC2计算单元，850GB存储，64位平台	1.2美元/小时	1.44美元/小时
	四倍超大	68.4GB内存，26个EC2计算单元，1690GB存储，64位平台	2.4美元/小时	2.88美元/小时

除了以上这种按小时的快速实例计费方案，Amazon公司还提供了其他可选方案，比如包年和竞拍方案。采用包年方案的用户只需一次性支付一定数额的费用，即可以相对更低的小时资费获得案例一的一年或者三年的使用权，该方案适合明确而稳健的业务需求；采用竞拍方案的用户能够以极低的价格拍得AWS在系统低负载时的实例，该方案适合对成本要求严格但对可用性要求较低的业务需求，或者离线处理的非实时性应用。

综上所述，云计算提供的这种在类型和时间上更加细粒度、在租期和要约上更加灵活的计费模型将有助于用户根据自身业务需求的特点来进行因地制宜的选择，达到减少初期IT投资的目标。

1.2.5 降低运营成本

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，云平台为用户提供了灵活的业务管理和便捷的服务。

1.2.6 产生新创价值

作为一种革命性的信息产业浪潮，云计算能够形成新的业务价值链，促进跨领域的创新协作，从而产生更高的价值。在以云计算推动的新价值链产生过程中，能够创造更多的就业机会，产生更多的新兴服务，建立新兴产业。

以我国台湾地区为例，随着2010年4月“云计算产业发展方案”的推出，一项跨度5年、投资240亿元新台币的规划进入人们的视野，预计该规划将在2年内改变千万民众的生活，创造5万个就业岗位，并最终实现1万亿元新台币的产值。台湾地区计划整合其在信息

技术、工程制造和基础设施上的优势，通过推动云计算来实现产业升级与节能减排的目标。“云计算产业发展方案”将由5大组成部分全方位推动，包括发挥台湾当地整体施政效益，提升该地区运作效能；提升民众生活水平；提升硬件附加价值；带动产业投资，加速产业转型；加强基础研究与产业科技研发等。

可见，云计算的革新虽然来自于IT行业，却将对诸如行政、教育、医疗等各个行业产生深远的影响。例如在中国台湾，2年后纸发票就有可能被全面取代，未来学生将连书包也不用带，市民看病也有统一的电子病历。云计算帮助实现了信息的整合与快速获取，将流程化繁为简，改变了人们的生活形态，并在这种新的形态中孕育了创新价值。

1.3 云计算产生的原动力

在1.2.6节，我们介绍了云计算的本质优势。实际上，早在1966年，D. F. Parkhill就在其经典的《计算机效用事业的挑战》（The Challenge of the Computer Utility）一书中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，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。

在过去的三十年中，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以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1.7所示。

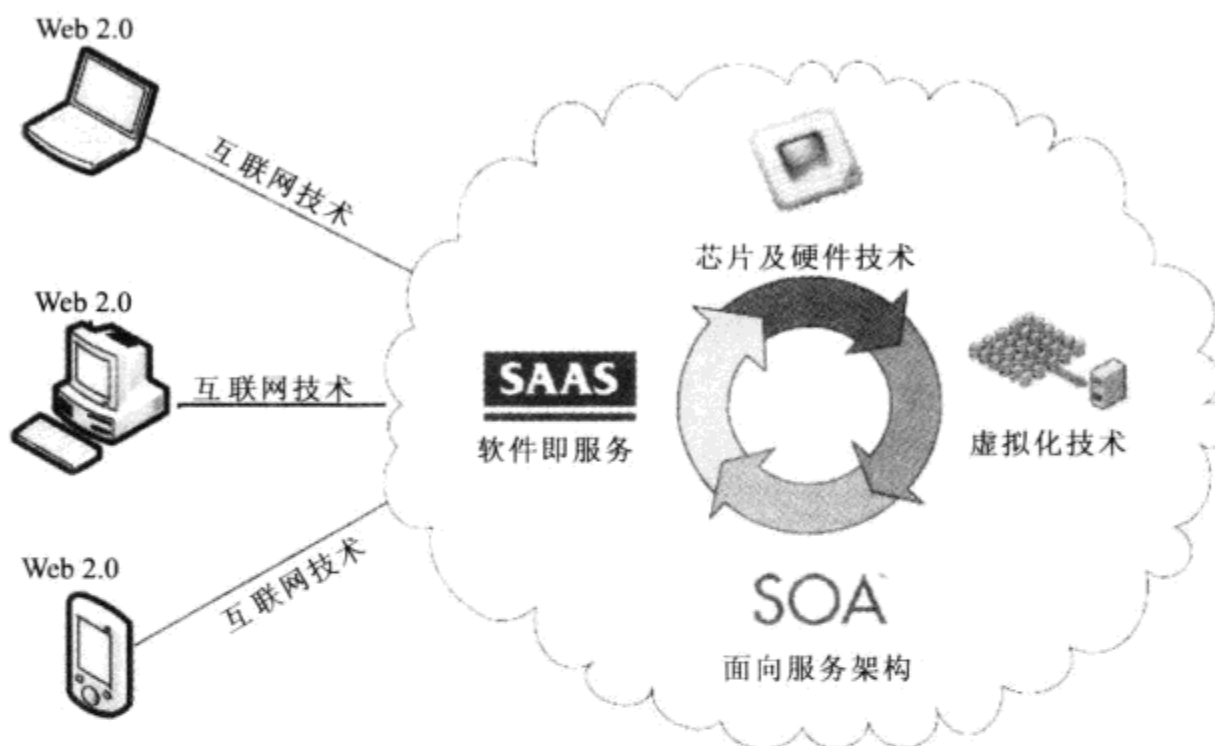


图1.7 云计算产生的原动力

(1) 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。

(2) 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。

(3) 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。

(4) 软件即服务模式流行，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。

(5) 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。

(6) 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。

1.3.1 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。

1.3.2 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。

1.3.3 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。

1.3.4 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，而改向服务提供商租用软件，且无须对软件进行维护，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，企业无须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。

1.3.5 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，互联网已成为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和LTE，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地可用了。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。

1.3.6 Web 2.0技术

Web 2.0浪潮推动了Web的功能创新性、信息共享程度和用户使用体验的长足进步。今天，它已经成为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使用感受。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和非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。

1.4 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本质优势，总结了云计算产生的原动力。变革的到来已势不可当。如果我们将视野放宽，把当下所处的时代定位于技术与经济发展的历史长河中，就能够更加清晰地理解云计算所带来的变革与机遇。Carlota Perez在其著作《技术革命与金融资本》（Technology Revolutions and Financial Capital）中梳理了从18世纪工业革命起到当今信息时代近250年的人类社会发展史，回顾了五次由于基础性的技术革命而主导的社会更迭，归纳了在这些更迭中由技术引致的从萌芽到高速发展、躁动到泡沫破裂、重建到最终成熟的循环规律。当前，我们正处于由IT技术主导的信息与通信时代，经历了.com泡沫的破裂，目睹了金融危机的爆发，社会的技术与经济正经历着秩序的重建。在这个过程中，虽然不再有虚幻的传奇和狂躁，但技术的沉淀和演进将越走越实，其中蕴藏的机遇和财富将越来越广。

1.4.1 大势所趋的转型

变革的背后往往蕴藏着规律，云计算的到来也不例外。它并不是随机的、意外的，而是当科学技术和社会经济发展到一定程度时必然发生的。在这里，我们将计算技术的发展与社会经济的发展做一个类比（如图1.8所示），便不难探究出其中的规律。

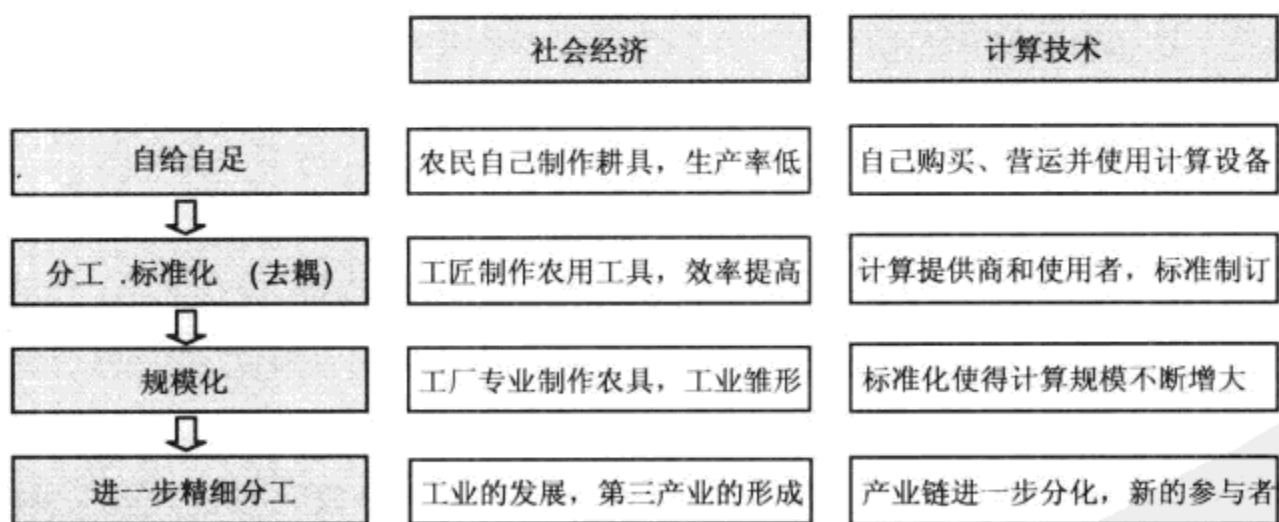


图1.8 社会经济与计算技术发展的规律

在农耕时代，家家户户过着“采菊东篱下，悠然见南山”般的生活，自己制作工具，自己耕种收获，看起来惬意，但只是低效地满足了有限的基本生活需求。随着铁制工具的

使用和生产技术的进步，农业劳动生产率获得提高，也使农具的生产变得多样而复杂，不能再由一家一户独立进行了。在这个背景下出现了人类历史上的“第二次社会大分工”，由工匠进行专业农具生产，由农民专门从事农业活动。随着社会生产率的进一步提高，工具的生产逐渐标准化和规模化，手工作坊的规模越来越大，农业生产的能力也越来越大，简单的交换已无法容纳社会经济发展的需求。在这个背景下，商人出现了，完成了“第三次社会大分工”。随着分工的不断精细化，每个人和社会整体的效率与效益都获得了提高，人类社会逐渐从荒蛮走向文明。

反观近百年来计算技术的发展，我们不难从中发现与社会经济相似的规律。在早期的大型机时代，公司或机构自己购买、营运并使用计算设备。随着半导体、网络 and 软件技术的综合发展，计算技术进入个人电脑时代。互联网上出现了多种多样的服务提供商，崭新的业务模式层出不穷。在这个过程中，提供商通过制定标准来巩固自己的专业地位与业务规模。这样的专业化与规模化不断深入，伴随着技术的成熟，IT服务将逐渐变得如水和电一样，可以通过无处不在的互联网随处获得。这就是云计算，顺应着历史发展的脉络在我们这个时代诞生。当下，仅是一个开始，随着IT生态系统的进一步精细分工，云计算将孕育出新兴的产业链。新的参与者将加入，旧的参与者必须相机而变。

1.4.2 新兴的产业链

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。

云计算产业结构中的角色如图1.9所示。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。可见，位于产业链中游的云提供商、云服务提供商和应用提供商从事着与云计算直接相关的业务活动，我们将他们统称为云计算提供商。

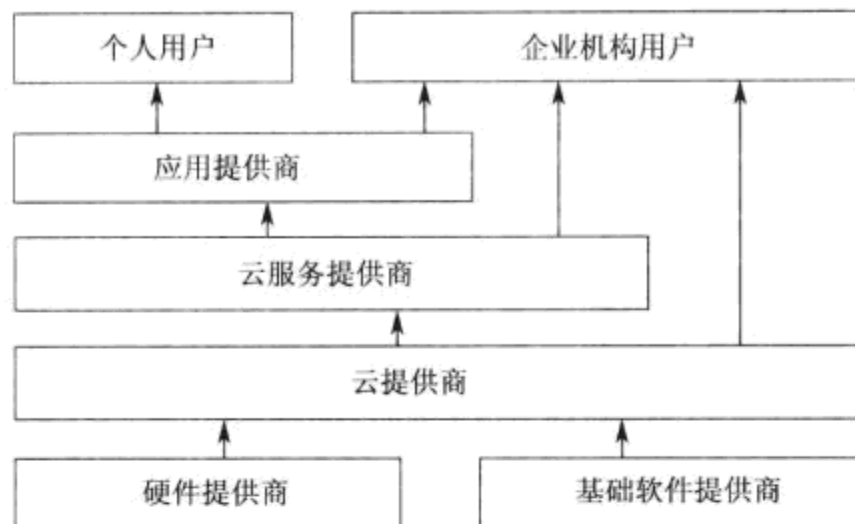


图1.9 云计算产业结构中的角色

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。

1.4.2.1 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。

1.4.2.2 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。

1.4.2.3 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。

1. 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。

2. 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。

3. 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。

1.4.2.4 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，就能够在云计算产业中获得成功。

1.4.2.5 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。

(1) 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。

(2) 由于云平台由专人维护，应用提供商也省去了维护费用。

(3) 云计算中所有的资源都按照具体情况付费，从而避免了传统方式中资源空闲所造成的浪费。

(4) 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。

1.4.2.6 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。

1.4.2.7 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，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，那些对安全性和可靠性要求很高的企业和机构，也可以选择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。

1.5 小结

本章从四个典型案例出发，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本质优势，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本质优势。然后，为读者解析了云计算产生的原动力。最后，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。

本章作为本书的引子，概述了云计算。在接下来的第2章将介绍几个典型行业中运用云计算的不同实践；在第3章，将按步骤介绍云计算的实施；在第4章，将深入技术层面，分析云计算的架构；在第5章，将介绍云环境构建的关键技术；在第6章，将介绍云计算的最新业界动态。



第 2 章

云计算的行业实践

- 2.1 概述
- 2.2 云计算在公共服务行业的应用
- 2.3 云计算在银行业的应用
- 2.4 云计算在电信行业的应用
- 2.5 云计算在物流行业的应用
- 2.6 云计算在医疗行业的应用
- 2.7 云计算在制造行业的应用
- 2.8 云计算在教育科研领域的应用
- 2.9 小结

进行云计算的行业实践分析，尤其是针对富有特色的中国市场，有助于我们更加深入地理解云计算。从已有的大量实践经验来看，创造利润和降低成本是将云计算应用于行业的两个最重要的主题。

从创造利润的角度看，企业必须通过开拓新市场和巩固优势市场等方式来获取新利润，满足资本需求。在产业链整合和混业经营愈加深入的背景下，企业必须整合上下游资源，打造核心竞争力，来获得新利润。云计算具有开放、整合的特点，能够适应企业的这种需求。从降低成本的角度看，企业对内必须降低运营成本。根据市场调研机构Forrester的报告，全球有超过40%的企业开始缩减IT开支。在这些开支中，用于维护硬件系统的部分约占80%。云计算资源共享的特征能有效降低投资成本，动态响应的特征能灵活适应管理需求、降低运营成本。

可见，云计算为身处各行各业的企业和机构带来了创造利润和降低成本的机会。但是，由于行业的差异性，云计算的实践也不尽相同。在这一章，我们将为读者剖析云计算在各个行业的不同实践。

2.1 概述

对于身处不同行业的企业和机构，在具体落地云计算时，在战略上需要注意以下几个层面。

1. 业务创新层面

云计算对业务的创新，业务是对象，创新是方向，云计算是手段。这里的重点要放在业务创新上，也就是一定要在理解业务现状和预期结果后，再考虑云计算的技术实现手段。云计算是实现特定业务目标的中介手段。在得到企业决策层的全力支持之后，云计算将成为推动业务创新的主要手段之一。

2. 技术创新层面

云计算涉及各项IT技术，云计算引领的发展趋势也为各种IT技术提供了广阔的创新空间。因此在技术层面，一部分厂商，比如运营商和互联网企业，需要加大对基础技术研究的投资力度，这一点从Google和IBM管理超过几十万台服务器的具体实践中就可以得到清

晰的诠释；另一部分厂商，比如跨平台虚拟化的硬件和软件厂商，需要从更大规模虚拟化的角度，升级老产品并研发全新的产品；对于系统集成商、负责业务建模的咨询供应商，需要深入研究云计算带来的变革和具体落地方案。

3. 风险控制层面

在开创新业务、引入新技术时，风险是应该被放在第一位考虑的问题。在当今的中国，超过30年的高速增长和对未来激动人心的预期都孕育着各种各样的机会，而如何识别、控制这些机会中潜在的风险则成为一道难题。首先，在具体实践云计算时，我们需要深入了解技术本质，细致定义风险，做好云计算落地时的风险控制。其次，由于现实市场的接受程度有限，云计算咨询服务商、集成商和设备供应商，以及使用云计算的用户要周详考虑云计算实践的困难和风险。

在综合考虑业务、技术、风险这三个层面后，企业需要根据自身IT资源和业务类型来选择合适的云计算环境，以便创造出最大的商业价值。

根据不同的IT资源基础，企业可以选择私有云、公有云或者混合云。对于规模较大、IT资源比较完善的企业，可以在自有的数据中心里搭建私有云；对于缺乏IT资源的中小型企业，可以利用公有基础设施云或平台云来创建自己的应用，也可以直接订阅公有应用云提供的服务；对于具有一定规模IT资源但并不充分的企业，可以通过应用和数据集成工具来搭建混合云，以实现了对现有资源的充分利用。

对于不同的业务类型，企业应选择相应的解决方案。各个行业的业务种类繁多，其中一些已经被验证能够运行在云中，而另外一些还有待进一步考察。目前，业界已经形成了一系列为不同业务量身定制的解决方案，可以应用于诸如开发测试、商业智能、桌面环境、存储服务等多个领域。无论是哪种业务迁移到云中，企业都需要根据自己的需求和应用的具体情况来进行综合的评估和决策。

当企业对自身IT资源和业务类型的判断达成共识以后，就可以将目标业务向云计算环境迁移了。这个迁移过程主要包括以下三个步骤

(1) 调研市场上可选的云计算服务提供商，这需要企业既深刻理解自身的需求，又广泛了解外面的世界；

(2) 深入考察候选者并最终选定提供商，这需要企业根据需求排除不满足条件的云

计算提供商，并对保留下来的候选者进行深入的评测和逐步的筛选，最终选定通过验证的提供商；

(3) 将业务部署到云计算环境，这需要企业将代码、应用和数据等内容向云中迁移，如果前两个步骤完成的效果好，最后的迁移过程将是平缓而快速的，企业将很快过渡到基于云计算的业务中来，并基于此创造新的业务和价值。

为了更好地帮助企业决策层制定合适的云计算战略，本章将从公共服务、银行、电信、物流、医疗、制造、互联网和教育这八大行业入手，深入探讨云计算给这些行业及产业链带来的机遇与挑战。

2.2 云计算在公共服务行业的应用

在推动云计算产业发展的进程中，公共服务，尤其是政府所发挥的作用越来越重要。

首先，公共服务业本身是云计算应用的潜在用户。比如，各城市建立的软件园需要借助云计算完成产业链升级；又如，像统计、气象、地震等需要高性能计算和高容量存储的公共服务部门也需要借助云计算减低成本并方便管理。

其次，来自政府的需求往往带有早期扶持的性质，有利于新产业形成初期规模，积累成功案例，提升产业信用。政府部门通过建立标杆性的云计算服务平台，形成良好的示范效应，带动产业生态环境的发展。

最后，也是最重要的是：政府制定政策的杠杆和指引作用。云计算是对整个IT行业涉及的各种厂商和服务供应商所采用的技术和业务模式的创新，是一个需要持之以恒改进的过程。政府相关政策，如积极制定针对相关从业者的优惠措施包括提供企业补助、低息贷款和减税等，又如制定云计算国家级、行业级的标准等，对云计算的应用均能起到极具意义的推动效果。

下面分别以政府部门及软件园为例，讲述云计算在公共服务行业的应用。首先看看各国政府在云计算的实践。各国政府都非常重视云计算的发展，很多国家业已制订出了相关的计划。在美国，云计算已在政府内部实质运作。2009年9月，美国总统奥巴马宣布了一项长期性的云计算政策，希望以此来压缩公共管理部门长期居高不下的支出。随后，联邦

政府启动了Apps.gov，通过它来展示并提供得到政府认可的云计算应用，帮助各级行政部门更好地接纳云计算的理念。

在韩国，知识经济部、通信委员会、行政安全部联合公布了《政府发展云计算综合规划》。三部门计划从2010年起至2014年5年间，在四大领域、十大详细课题方面共投入6000万亿韩元。这项投资主要包括：“公共部门率先构建”（4488亿韩元），用以建设政府综合数据中心及跨政府云平台；“构建民间云服务基础平台”（514亿韩元），用以建设八大试点项目及测试平台，“云技术开发”（582亿韩元），用以开发云计算的原创技术；“营造发展环境”（48亿韩元），用以完善相关法律法规和制度，构建安全及标准化体系。

日本政府正在建设名为“Kasumigaseki Cloud”的云计算环境，为公共管理部门的各种应用提供一个安全、统一的基础设施，使这些部门之间能够更好的协同。在这个系统中，一些支持性功能，如工资、审计和个人信息管理等业务将运行在私有云中；而一些面对公众的服务则将被放在公有云中。除此之外，日本超过1800个的地方政府也拥有属于自己的IT资源，该系统将对这些资源进行分组管理，每个组拥有自己的私有云，这些私有云与Kasumigaseki Cloud连接在一起，形成混合云。

我国政府决策层高度重视云计算，胡锦涛总书记在今年的两院院士大会中提到，云计算等领域的快速发展为个性化制造和服务创新提供了有力工具和环境，人依靠机器生产产品变成机器围绕人生产产品成为可能，个性化制造和规模化协同创新有机结合将成为重要的生产方式。在实践方面，各级政府在云计算方面也开始了相应的尝试，其中北京、上海和无锡等地已经制定了相应的政策或示范项目，来推动云计算产业的发展。在北京市政府的领导下，北京市计算中心正在推进“北京云”计划，它的初衷是为了解决北京市中小企业的计算需求，提供服务化的工业计算、生物计算和科学计算应用，服务类型包括Ansys、Fluent、Abaqus、BLAST、Gromacs等20余种。用户可采用基于Web网页或命令行的方式提交作业，实现对应用服务的定制。

上海市经济和信息化委员会发布了《上海推进云计算产业发展行动方案（2010-2012年）》，该方案为上海成为“亚太云计算中心”制定了几大发展目标：第一，在三年时间里培育十家年经营收入超亿元的云计算技术与服务企业、建设十个面向城市管理、产业发展、电子政务、中小企业服务等领域的云计算示范平台；第二，推动百家软件和信息服务业企业向云计算服务转型；第三，带动信息服务业新增经营收入千亿元、培养一千名行业尖端人才。无锡也建立了由政府资助的云计算中心——“盘古天地”，希望籍此激发当地软件外包服务的创新，提高该区域技术经济的增长。

综上所述，无论是出于节约开支、提高效率的考虑，还是出于提升区域竞争力的考虑，对于公共管理部门来说，云计算都是关系到国家信息产业发展的战略性课题。在我国，政府对云计算的支持，还主要停留在地方政府的层面，而在其他国家，尤其是美国，中央政府已经出面采取行动。美国政府对于新技术的应用态度很坚决，推进力度也很大，已取得了初步的成果。在即将到来的云计算时代，政策的不同是否会拉大中国与其他国家在云计算领域的差距，这个问题应当引起相关部门的足够重视。

云计算在公共服务行业的第二个典型应用是高新软件园区的产业升级。

近年来，我国各地都兴建了许多高新软件园。这些软件园提供给园区租户的服务通常比较相似，如场地租用、税收政策优惠和资金扶持等。虽然这些软件园在一定程度上推动了当地软件产业的发展，不过也存在着一系列的问题，比如服务内容有限，培训、咨询、融资、法务、市场推广等“软能力”薄弱，产业整体资源共享程度低、协同作用差等。此外，还有园区租户的基础软硬件建设投资重复、创新门槛高等问题。为保证园区长期发展，避免租户在税收优惠到期就更换园区等不稳定因素的出现，园区需要提供更好的服务来满足社会效益和经济效益的提升。对于租户，也就是园区内的软件企业而言，也面临着一些问题，比如市场拓展能力差、软件开发周期长，获取利润能力低，以及知识产权保护意识弱等。

为解决这些问题，基于云计算的软件服务技术创新与孵化平台方案获得了业界的普遍关注。从业务模式角度看，云计算平台方案主要包括如下内容，如图2.1所示。



图2.1 针对软件园的云计算平台方案

从图2.1中可以看到，该方案以软件园、园区租户、技术提供商三方的紧密合作为基础，软件园向租户提供运营平台、技术创新支持、激励政策等，租户在得到支持后可以快速发展，为软件园贡献利润，提升软件园形象，而技术提供商作为创新和咨询服务的提供者，帮助软件园和租户双方进行提升，同时，技术提供商也获得了宝贵的研究平台和实践机会。平台主要包括创新与孵化平台建设、业务咨询与发展规划、技术咨询与孵化、平台服务与运营、产业推广与配套政策几大环节，中心目标是为软件产业和中小软件企业提供孵化平台，通过几个环节的综合作用，以及三方的紧密合作，实现共同发展，形成多赢的局面。

在这种建设内容下，软件园区、园区租户（一般包括服务提供商和软件开发商）和租户的用户（软件最终用户）从云计算平台中将分别获得如下价值提升。

对于软件园而言，云计算平台具有五大价值。

- (1) 云计算可以扩大招商引资，对中小企业形成巨大的吸引效应；
- (2) 云计算平台有利于形成产业规模，铸造品牌；
- (3) 云计算平台可以使各方共享收益，成为创新利润增长点；
- (4) 云计算平台可以促进相关产业发展，形成产业链条和产业生态系统，促进产业升级；
- (5) 云计算平台还可以通过细分产业链条的方式创造出更多的就业机会。

对于中小软件企业租户来讲，云计算也同样给他们带来了机会。

- (1) 云计算平台可以降低创新成本、开创新的业务模式；
- (2) 云计算平台可以提高标准化程度与软件服务质量；
- (3) 云计算平台可以缩短周期，灵活应对市场变化；
- (4) 云计算平台可以建立平台集体品牌，以协助企业发展。借助政府公信力与园区服务供应商的品牌价值，软件企业通过平台的集体品牌从市场的角度更加容易拓展新的市场与客户群；
- (5) 云计算平台可以形成聚集效应，增大客户群；

- (6) 云计算平台可以帮助中小企业吸引和培养专业人才；
- (7) 云计算平台还能够提供较低的进入门槛和灵活的合作模式。

对最终用户而言，云计算平台可以提供两大便利。

- (1) 它提供了快捷低成本的SaaS服务；
- (2) 它可满足客户全面的需求并实现与之相关企业的信息共享与服务集成。

2.3 云计算在银行业的应用

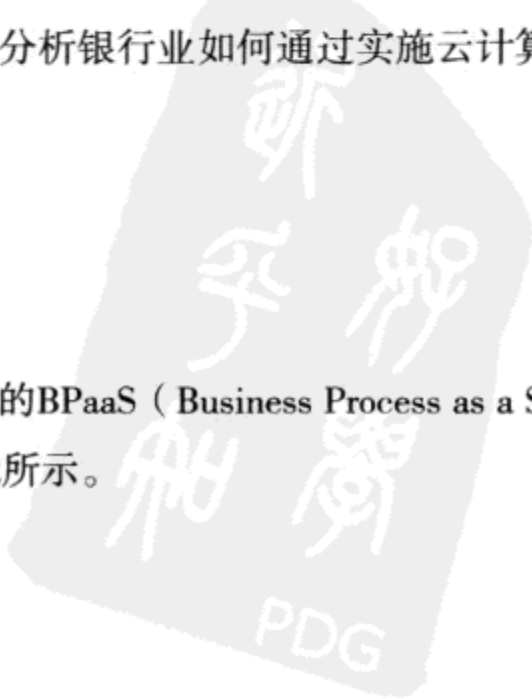
当前银行业的核心业务是存款和放贷，核心利润来源是赚取息差。简单来说，银行吸引储户存款，支付一定的比例利息给储户；银行放贷给企业，收取一定比例的利息。这两者之间的利息差，也就是“净利息收入”，是传统银行利润的主要来源。

虽然有国家相关政策的支持，但单靠净利息收入显然不足以进一步增加银行的竞争力和风险有效控制能力。由此，国有银行和商业银行纷纷把进一步创新利润的来源放在中间业务上。所谓中间业务，主要是以手续费及佣金为代表的非息差业务。从各大银行的财报来看，中间业务的普遍增长一定程度上维持了银行整体利润的增长。

云计算能够帮助银行进行业务创新，而业务创新的核心资源是：金融资本。因此对银行业而言，当新兴的云计算和传统的资本管理技术结合到一起时，创新的空间将十分可观。同时，银行需要在自身IT基础领域进行高额投入，这也为节约成本提供了空间。下面，我们将通过三个具体实例来分析银行业如何通过实施云计算来创造新利润、控制风险和节约成本。

2.3.1 中间业务创新

我们以某商业银行与某商场的BPaaS（Business Process as a Service）云计算实践为例说明银行中间业务的创新，如图2.2所示。



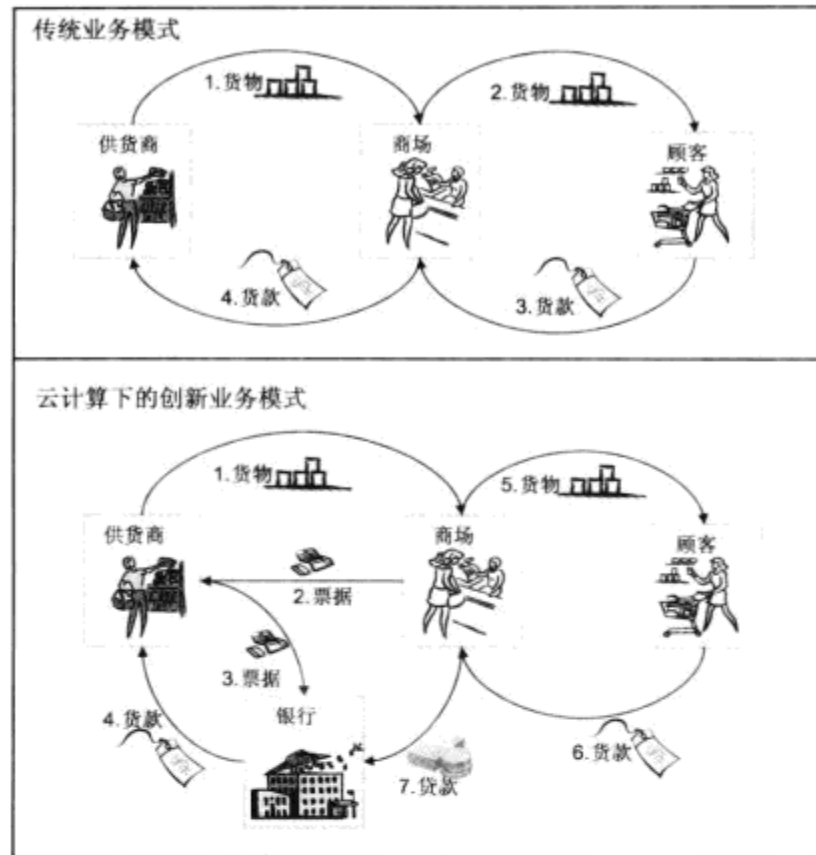


图2.2 云计算为银行业带来的中间业务创新

在图2.2上半部分所展示的业务流程图中，我们可以看到商场的传统业务模式：商场从供货商拿货、卖货，把部分货款还给供货商，赚取价差。从商场角度，如何保证卖的货比别的商场价格低，是在这种业务模式下的核心要义。降低所采购货物的成本，是业务的创新点所在。从供货商角度，如何尽快拿到回款是关键。我们可以用著名的杜邦财务公式来分析加快回款，也就是加快资金周转率的重要性。

杜邦公式：

$$\begin{aligned}
 \text{净资产收益率} &= \text{净利润} / \text{股东权益} \\
 &= (\text{净利润} / \text{销售收入}) \times (\text{销售收入} / \text{总资产}) \times \\
 &\quad (\text{总资产} / \text{股东权益}) \\
 &= \text{销售利润率} \times \text{总资产周转率} \times \text{财务杠杆比率}
 \end{aligned}$$

所谓净资产收益率，就是企业经营者要投入多少钱做生意，并能赚回多少钱的比例；这是企业最核心的经营要义，也是进行业务创新的核心所在。所谓销售利润率，简单理解就是毛利率。毛利率的提高非常困难，但如果要运作好一项业务，还可以考虑总资产周转率和财务杠杆比率。所谓总资产周转率，就是用365天除以从拿现金买原材料、生产、销售、并最后回款的天数。所谓财务杠杆比率，就是自有资金与能够吸引的

资金的比例。如果财务打理得好、业务经营得好，这个财务杠杆可以非常高。房地产行业之所以吸引很多资金的涌入，原因之一就是财务杠杆比非常高。

对于供货商，如果第一种情况是毛利30%，第二种情况是毛利20%，而第一种情况下资金周转率是4（也就三个月一个周期），第二种情况资金周转率是12（也就是1个月周转一次），那么在同样的资金杠杆比情况下，供货商的在第二种情况下的净资产收益率将是第一种情况的两倍，即： $(12 \times 20\%) / (4 \times 30\%) = 2.0$ 。换句话说：通过简单地提升资金周转率，将能100%地提升供货商的整体利率。

但是，商场为了提升自身的财务杠杆比率，通常都会压很多“头寸”，也就是通常让供货商在2个月以后才拿到货款，这就压缩了供货商的资金周转率。供货商在资金周转率低的前提下，为自身销售利润率，会降低原材料采购成本。因此，在这种产业链模式下，商场和供货商难于赚到合理利润，而顾客拿到的却是高价低质的产品。

基于BPaaS的云计算理念，以上业务模型可以引入由银行提供贷款，进行业务模式的创新。如图2.2的下半部分所展示的，在云计算下的创新业务模式中，供货商提供货物给商场后，可以从商场立刻拿到票据。之后，供货商找到银行，银行根据票据立刻付款。由于无需等待顾客购买货物的过程，供货商的资金周转率可以立刻从月为计量单位到以天为计量单位。

回顾上文的杜邦公式，供货商是非常愿意通过缴纳一定比例的手续费（也就是降低供货商毛利）的方式来提升资金周转率的。这部分的手续费，一部分作为商场的利润回报，吸引商场参与到这个BPaaS的业务创新中，另一部分则作为银行发放贷款的利润回报。商场还可以拿出一部的利润，做积分卡、促销卡等，吸引更多顾客到商场购物消费。这样，顾客也能享受到利益。

在这种业务模式下，是商场而不是供货商申请贷款。商场一般都有较大的固定资产，银行风险可控度相对较强。更重要的是，整个商场的进销存系统运行在公有云平台上，银行可以看到商场和供货商整个供应链的财务状况，使银行的风险进一步可控。可见，银行业的中间业务，借助云计算的智慧，实现了再次的创新，真正为银行创造了新利润。

2.3.2 核心业务创新

银行的传统核心业务是赚取息差，而风险控制则是最大的挑战。当前银行贷款发放对象主要是大中型企业，其中很大一部分是房地产商。由于国家持续对房地产市场进行宏

观调控，同时在金融危机发展仍存在未知因素等背景下，不良贷款出现的可能性加大。为此，银行普遍采取所谓的“逆周期监管”计划，即在实体经济下滑期、复苏期和调整期，管理层采取一系列比在实体经济上行期更为严格的金融监管政策和措施。这种愈加严厉的监管行为在一定程度上能缓解不良贷款率的上升幅度。但在本质上，银行必须通过业务创新，开辟风险更小的新增贷款业务市场。

从另一个角度看，处于发展初期的中小企业急需资金，但由于缺乏可抵押的资产和普遍较低的存活率，银行很难控制风险。同时，由于中小企业比较分散，银行服务成本高，难以保证息差利润。因此，一直以来，中小企业一直很难从银行贷款，拿到能决定企业生死的启动资金和周转资金。

为了进入这个几乎空白的市场，银行可以基于云计算的理念和技术，选择其中最具有需求、风险最可控的细分市场，例如从事B2C（Business to Customer）电子商务的中小企业。根据中国电子商务研究中心发布的《2010年（上）电子商务市场数据监测报告》，从事B2C销售的网店去年就已超过1000万家，并以每天5000家的速度在增长，同时预测2010年网购销售规模可能达到惊人的5000亿人民币。通过这些数据，我们可以毫不怀疑地看到网店这个巨大的中小企业市场。

云计算为银行业带来的核心业务创新如图2.3所示，在基于云计算的创新理念下，B2C电子商务平台运营商被整合进来，将直接解决银行与中小企业的贷款难题，实现优势互补，共同发展。

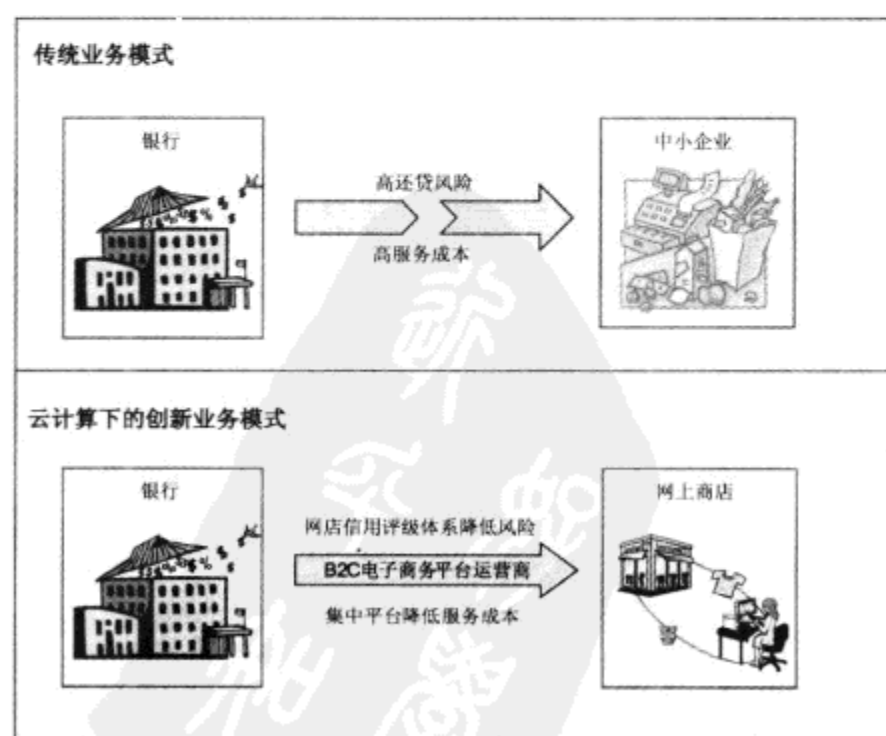


图2.3 云计算为银行业带来的核心业务创新

一方面，B2C电子商务平台运营商为解决网店本身的交易问题，花费了巨大的力气去解决信用评级体制问题，而这个信用评级体制恰恰就是银行最重要的风险控制措施——信用评估体系在电子商务领域的翻版。因此，基于电子商务运营商所运营平台的平台效应，银行可以集中处理物理上极其分散的中小企业，这从根本上解决了银行提供贷款服务成本高的问题。更重要的是，银行可以直接利用电子商务运营商的信用评级结果，作为重要的风险评估手段和依据，以控制贷款风险。

而另一方面，B2C平台运营商为了更好地支撑自身的发展，就要更好地服务于客户，也就是开网店的这些中小企业；如果能够帮助这些中小企业解决企业发展中最重要的资金问题，B2C平台运营商能更进一步凝聚中小企业对所运维平台的依赖性。因此，银行可以帮助B2C平台运营商提升服务质量，提高客户忠诚度。

借助云计算的先进理念，银行、电子商务中小企业和电子商务平台运营商均实现了多赢。而云计算的技术整合能力，能够让银行的信贷系统、电子商务中小企业的财务系统和电子商务平台运营商的信用评级系统三者进行互联互通，从而确保了业务创新模式的真正落地。

2.3.3 开发测试业务创新

一项金融业务的诞生，从对它的分析论证、设计规划到相关应用的开发测试、上线运行，都需要IT工作人员方方面面的配合。在这里，开发测试业务是云计算应用于银行业的切入点之一。在传统模式下，对于一个新的应用，业务开发部门需要将其所需的IT资源申请分解成多个工单，交给IT运维团队来完成配给。这种方式既给运维团队带来了繁重的工作量，又由于缺乏良好的规划，使服务器的利用率偏低。云计算能够很好地解决传统模式下的这些问题。通过访问开发测试云中的资源，使用者能够将工作所需的系统和中间件自动部署到虚拟化环境中。这样做既减轻了IT运维团队的工作量，又使数据中心资源的动态优化成为可能。

在银行业中应用开发测试云，必须考虑到行业自身的特殊性。

(1) 金融机构数据中心的规模大，单就小型机而言，大型银行的数据中心一般拥有几十台，其上运行着成百上千个虚拟服务器；

(2) 金融机构数据中心内的服务器种类多，包括大型机、小型机以及x86服务器，而

且可能来自不同的设备提供商；

(3) 为了保证上线的应用能够及时地被更新和调试，数据中心通常会保留它们的开发测试环境。因此，金融机构在将开发测试工作向云中迁移时要做到循序渐进，既要在构建云环境的过程中考虑系统的异构性问题，又要在运行云环境的过程中将遗留的应用系统纳入到管理范围之内。

另外，对于银行相对不太核心的网站业务、增值服务等业务的开发和测试，可以尝试使用公有的开发测试云，例如IBM的Development & Test Cloud。

2.4 云计算在电信行业的应用

云计算需要对IT基础技术进行深入的研究和投入。电信运营商在对这些基础技术的实际应用、试验环境和企业资本方面都具有独到的优势，在云计算领域可大有作为。由于国内三大运营商业务发展状况和拥有核心资源的不同，它们对待云计算也有不同的态度和行动。

中国移动拥有最多的用户数量，其战略重点放在提升每用户平均收入值上，因此需借助云计算的先进理念和技术进行多种增值业务的开拓。中国移动核心的“大云”产品包括五部分：并行数据挖掘工具、分布式海量数据仓库、弹性计算系统、云存储系统和MapReduce并行计算执行环境。中国移动在2007年3月确定了大云的研究方向，同年10月进行了首次扩容，建立了256个节点的规模试验环境；12月进行了再次扩容，节点数量增加到了1024个。可见，中国移动在云计算领域的投入力度很大。

中国联通尚未进一步披露关于“互联云”实际性的规划和产品成果，但有报道认为联通将会以IDC（互联网数据中心）转型为发力点。中国电信同样也未进一步披露关于“星云”的信息；但从前期试点城市的实验内容看，有报道认为也是以IDC转型为发力点。

但是不论哪家运营商，云计算在业务支撑系统、增值业务系统、企业内部IT管理系统、测试和离线运行环境，以及IDC均存在相关的机会。此外，云计算还能够促成运营商的创新业务模式，如移动支付等。云计算在电信行业的机会如图2.4所示，云计算能够为以上各个领域带来共同的价值，包括通过资源的虚拟化管理和资源共享，提高

资源利用率，从而降低投入，提升利润率；以及通过资源的自动化和精细化管理，降低管理成本，从而降低运营成本。针对各个领域，云计算能够为它们带来该领域中特有的如下价值。

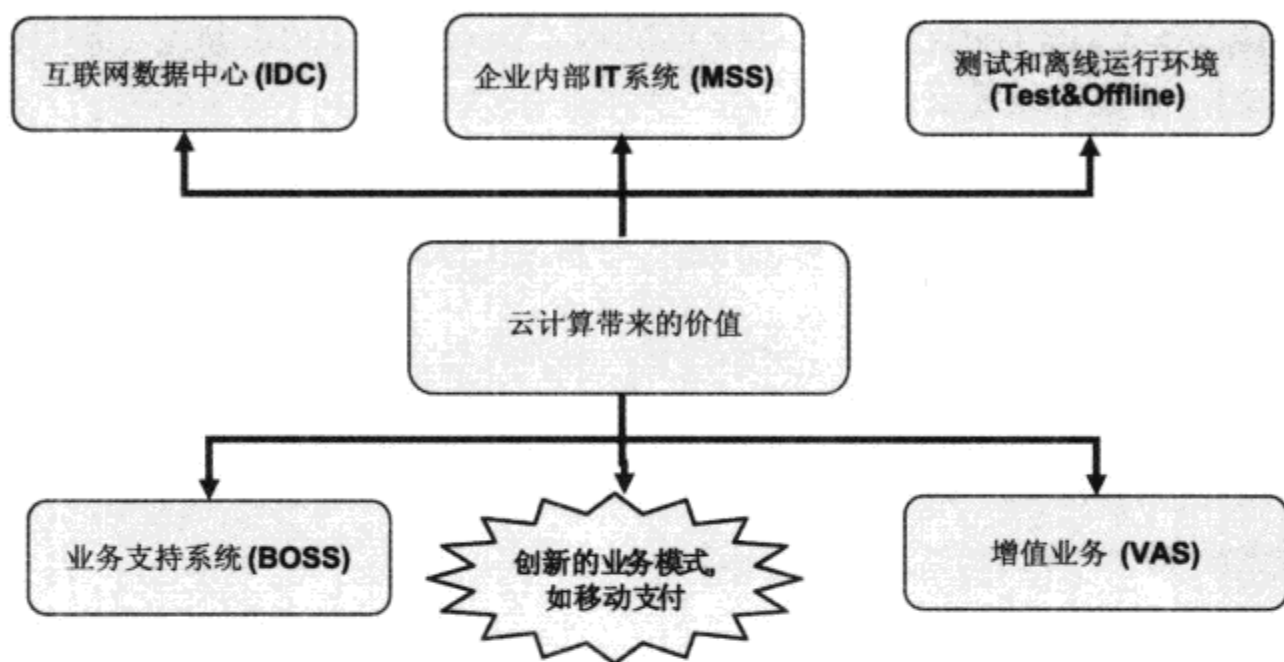


图2.4 云计算在电信行业的机会

- ▶ 对于业务支持系统，云计算能够为业务高峰期处理的要求分配更多资源，提升客户满意度；集中化的动态部署，例如网站、外网门户、数据集市、接口服务等边缘化应用，提高集中管理水平。
- ▶ 对于增值业务系统，云计算能够根据市场反应和用户反馈，动态调整增值业务对于资源的使用，加大为企业带来更多营收的增值业务投入；及时释放和调整运营周期短、市场反应冷淡的增值业务对于资源的占用，提高资源利用率；通过降低合作伙伴的进入技术门槛，极大地丰富产品组合，增加营收。
- ▶ 对于企业内部IT管理系统，云计算能够为全企业提供统一的IT基础环境，从而方便集中化管理，减少重复建设，节省投资。
- ▶ 对于测试和离线运行环境，云计算能够依据不同项目的需求进行定制化搭建，提高资源利用率，降低运营维护工作量及成本；避免纯手工搭建测试平台的费时费力并且容易出错等缺点，把可以离线运行的业务转移到测试环境中运行，减少现有业务系统的运行压力。
- ▶ 对于互联网数据中心，云计算能够提升业务创新和上线的效率，帮助快速部署业

务，提升竞争力，增加营收。

下面将通过一个IDC的云计算实践实例，来帮助读者理解云计算在电信行业的应用。

2000年之前，中国IDC市场曾以每年50%以上的超高速增长。这种超高速增长源于互联网泡沫以及人们的投机心理。2000年互联网泡沫破灭后，IDC业务进入蛰伏期。2002年以来，中国IDC业务再次迅猛增长。这一阶段的增长则更多地带有理性特征，其增长基础是短信、网游、语音以及视频宽带业务的日益火爆。各种类型的服务/内容提供商兴起，成为IDC业务的重要客户群。IDC业务迈向第二轮高速增长期。

早期的IDC业务范围主要包括：网站托管、服务器托管、高速接入、应用托管、企业网站建设、管理和维护。随着业务发展和客户增长，IDC开始推出负载均衡及集群服务、Web缓存服务、VPN服务、网络存储服务和网络安全服务等增值业务。到目前，以服务器托管为代表的基础业务依然是市场的主体。

目前，电信运营商都不约而同地加快了数据中心的改造。一方面是因为IDC的盈利能力都不强，迫切需要改变；另一方面，IDC似乎更直接地需要云计算的大规模、动态、虚拟化等的技术特点。如表2.1所示，某运营商的一个省级IDC遇到如下的实际问题。

表2.1 某IDC遇到的实际问题

业务类型	核心资源	主要客户及应用	发展瓶颈
增值业务	应用	<ul style="list-style-type: none"> · 企业客户之辅助业务 · 小规模商业之辅助业务 · 个人客户之资讯和娱乐 	<ul style="list-style-type: none"> · 附加价值低 · 客户业务关键性低 · 客户忠诚度低 · 竞争激烈，进入门槛低
机房租用	机房基础设施	<ul style="list-style-type: none"> · 小规模商业之互联网业务 	<ul style="list-style-type: none"> · 附加价值低 - 收入和利润低 · 客户业务关键性低 · 客户忠诚度低 · 竞争激烈 - 进入门槛相对偏低
带宽租用	电信基础设施	<ul style="list-style-type: none"> · 企业客户之核心业务 · 小规模商业之互联网业务 · 个人客户之互联网访问 	<ul style="list-style-type: none"> · 竞争激烈 · 企业私有网 · 同业竞争 · 缺乏新的带宽使用增长点

在这种背景下，该运营商在深入研究云计算的技术特点后，借力业界领先的某服务供应商进行了针对IDC业务、IDC IT基础架构和IDC运维等全方位的咨询工作。在IDC业务咨询中，该服务提供商帮助运营商确定调研范围，研究其现有的IDC业务并进行客户访

谈，基于此设计IDC业务体系。在IDC IT基础架构咨询中，该服务提供商调研运营商原有IDC的网络、主机、存储、安全、容灾等基础架构，给出五星级IDC机房建议以及相关的设计细则。在IDC运维咨询中，该服务提供商调研运营商现有的运维体系和现状，分析人员组织架构，给出运维改进建议和省/地市二级运维体系建议。

通过以上的咨询分析工作，该运营商发现，如果通过云计算的方式进行资源配置、管理和出租，从投资回报（ROI）分析可得出：出租2000个Core的虚拟机，即可实现云计算业务的盈亏平衡。但问题是如何吸引更多的租户将自身生产系统运行在运营商IDC的云平台之上。

解决这个问题的关键除了进行营销推广外，更重要的是在当地的大型用户中建立起样本工程。于是，在服务供应商的帮助下，运营商对当地大型客户进行了仔细斟酌，最后选定一家当地最大的、已上市的、正在进行全球化的企业重点突破。服务供应商借助于与该企业的良好关系，对该企业所有应用做了负载分析、风险分析和迁移分析，选择出其中23个非核心应用。通过严格的投资回报分析，认为这些应用迁移到云平台上可以帮助该企业每年节省40%以上的成本。从风险角度看，由于选择的应用是非核心应用，风险相对可控。同时，服务供应商还在自身搭建的云平台上对企业的应用进行了实际的概念可行性验证（POC）和技术可行性验证验证（POT），进一步证实风险的可控性。该企业很快与运营商签订了框架协议，运营商打包售出了1200个Core的虚拟机。于是，该运营商决定投资进行IDC改造，并预计在2011年正式对外提供云IDC服务。

2.5 云计算在物流行业的应用

现代物流业已经成为一个重要而复杂的产业。其典型产业链包括如下几大部分。

- （1）货主、发运人和收货人：包括流通企业、生产企业等；
- （2）基本运输能力供应商：包括航运、空运、铁路、公路、港口、堆场、仓储等；
- （3）物流服务供应商：代理、第三方物流等；
- （4）金融服务供应商：包括银行、保险公司等；

(5) 连通服务供应商：包括国际供应链、相关企业平台等；

(6) 政府监管部门：包括海关、检验检疫、税收等。

随着信息技术的发展和全球化进程的推进，物流企业纷纷采用IT技术来提升自己在全球范围内的竞争力。但即使如此，物流供应链仍然存在很多问题。有数据显示：由于物流供应链的低效，消费品和零售业每年损失约400亿美元，相当于其销售额的3.5%。为改变这种现状，进一步提升物流行业的价值，物流行业从业者纷纷从第三方物流开始转型为第四方物流。第三方物流与第四方物流最大的区别在于：服务的内容从过去只是单纯的提供仓储、运输服务转变成整合的供应链解决方案，其服务范围包括现金流、库存控制、货物跟踪、异常反馈、运输装载等。也就是说，物流产业要从独立的运作体系转型为区域性的整合，并以此提供快速、准确、准时、即时的信息，为客户实现完整的物流过程。

这种产业链的升级，使得物流企业需要更具有弹性与灵活性，通过实时、动态的配置，让资源得到更合理、更充分的分配与利用，进而发挥地域优势，降低体系成本，达到快速响应客户，提高竞争力的目标。而快速、灵活、大规模的整合，正是云计算的优势。在全球化的业务运作中，也唯有通过云计算，才能够使企业在任何时间、任何地点使用动态的IT资源。物流服务云计算平台的业务模式如图2.5所示。

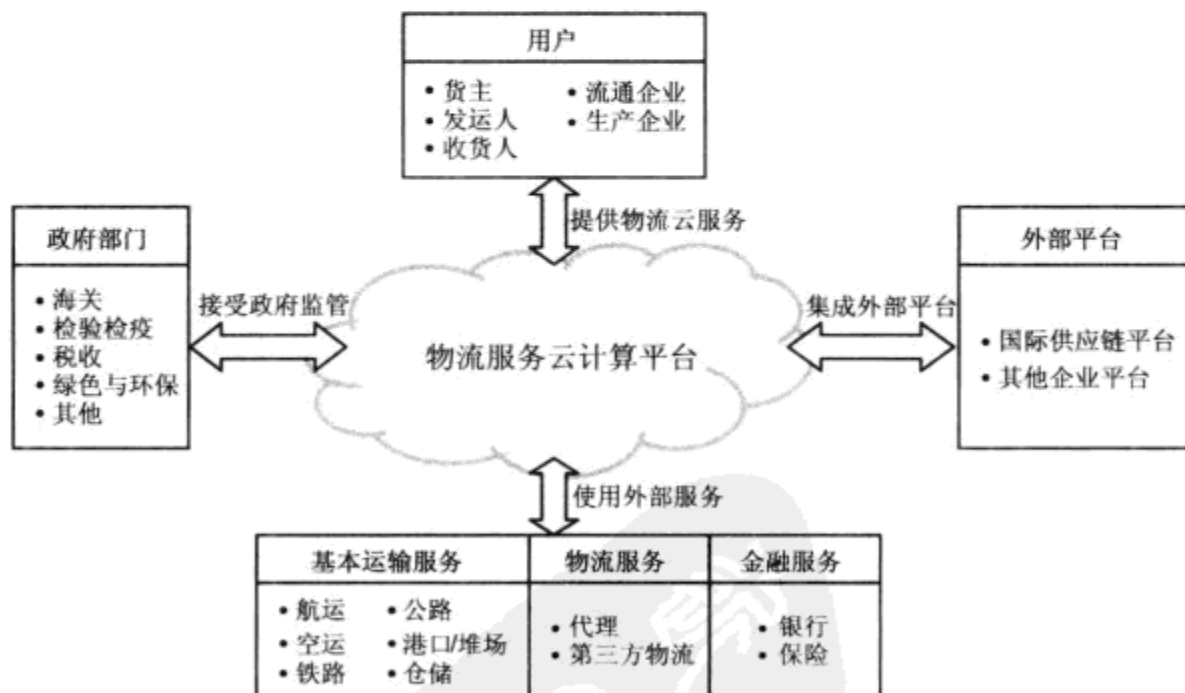


图2.5 物流服务云计算平台的业务模式

从图2.5中我们可以看到物流企业最常遇到的困难，也就是“整合问题”是如何被解决的。为了保证发货端到收货端的顺畅，物流公司的集散中心要足够多。以中国某大型物流

公司为例，其在我国的北方、西南、华南、华中等地区的集散中心覆盖完整，但在西部内陆地区，如西安、兰州、拉萨等城市的集散中心偏少，造成物流在运送节点上的中断。为兼顾成本和服务质量，该物流公司采用与当地中小型物流企业合作的模式。为了给客户提供的优良服务，掌握货物的递送情况，该物流公司利用云计算，打造上、中、下游的共享平台，建立了物流业的公有云，让中小型物流业者不必再自建系统，只需登录该物流公司打造的公有云平台，即可进行订单、仓储、配送的管理。

云计算的特性在于通过网络和共享的IT资源，用最少的管理成本，达到最快的资源配置和最大的效能。物流云的公有云平台，对该物流公司及其合作伙伴都有益处。对该物流公司来说，它无须付出高额的费用来自建集散中心，每年节省下约上千万元人民币的配送与库存成本；对于中小型的物流企业来说，他们无须投资自有IT系统，只需通过网络和浏览器就能进入该全国性物流公司的公有云平台来处理与扩展业务。

提高物流公司的运输效率异常重要。但以往由于信息流通得不畅或不及时，物流公司的空车率往往较高，影响了整体效率。通过物流云，物流公司能够实时了解物流网上的货运情况，直接安排途中接货，减少空车行驶情况的发生。另外，物流云还能够解决物流企业在仓储与运输上的衔接问题，整合孤立系统，提供多元化的服务，例如物流系统与银行现金流系统、天气预报系统和交通信息系统的对接等。

另外，物流系统与银行现金流系统对接之后，银行可以根据物流云上厂商的库存状态和运输的情形，评估其所提供给厂商周转的信贷利率。如果某厂商的库存管理情况不良，银行可以及早在资金方面有所处理，降低银行的投资风险。物流系统与天气预报系统和交通信息网络相对接之后，物流企业能够进行有效的路径规划。在途司机可以通过随身装备及时发现前方即将出现的恶劣天气，更改物流路线。

综上所述，物流云帮助物流企业达到的不仅是运输、仓储的整合，更是物流、资金、贸易、研发等的整合，是一项集上、中、下游的全面整合。

2.6 云计算在医疗行业的应用

据报道，中国某一线城市，每年挂专家号的人次在一亿以上，而该城市每年可接待

专家问诊的能力在一百万左右。实际上，大量挂专家号的患者，很多只是感冒之类的小症状，完全不必在大型专科或综合性医院就医。资源调配的不合理严重影响了医疗行业的整体效率，也直接导致了医疗质量难以保证、地区之间参差不齐及医患纠纷增多等状况。

这种现象的产生与IT系统的建设模式也有深刻联系。在传统的医疗系统中，服务器、网络和存储等IT基础设施往往是分散而隔离的，由不同的医疗机构或者同一医疗机构的不同部门单独维护和使用。这些分离的系统无法做到对信息的有效共享和对医疗系统的统筹管理。而云计算的出现为实现医疗信息系统的联合优化和动态管理提供了可能。云计算可以将这些分散的系统整合在一起，形成统一的医疗信息基础设施，提供类型多样的健康管理应用，为每一个人制订个性化的方案。此外，在生物医学和个性药物的研究过程中会涉及大量数据处理和计算。云计算节约资源、便利管理的特性也将提高这些领域的研究效率。

为此，很多国家的政府都在考虑基于云计算的医疗行业解决方案。比如，美国的医疗计划有一个雄心勃勃的目标，通过云计算改造现有的医疗系统，让每个人都能在学校、图书馆等公共场所连接到全美的医院，查询最新的医疗信息。丹麦政府计划通过云计算建立全国性的医疗体系，从而改善该国药品管理局的工作流程，并将优化的流程推广至药商甚至全丹麦的医药行业。

在我国，温家宝总理在《政府工作报告》中指出“医疗体制改革要实现人人享有基本医疗卫生服务”。新医改的第十四条也明确提到：“加快医疗卫生信息系统的建设，以建立居民健康档案为重点，构建乡村和社区卫生信息网络平台”。要达到这个目标，必须建立新的医疗系统和平台。现在政府正在全力推广以电子病历为先导的智能医疗系统，要对医疗行业中的海量数据进行存储、整合和管理，满足远程医疗的实时性要求。云计算是建立智能医疗系统的理想解决方案，通过将电子健康档案和云计算平台融合在一起，每个人的健康记录和病历能够被完整地记录和保存下来，在合适的时候为医疗机构、主管部门、保险机构和科研单位所使用。

同时，一些知名的医疗研究机构也开始使用“健康云”。美国哈佛医学院是最早部署和使用云计算平台的医疗机构之一，它所建立的私有“医疗云”已经成为其在日常医疗和研究工作中不可或缺的一部分。哈佛医学院的研究人员和工作室分布在波士顿的多个地点，其中有6个基础研究实验室，50个门诊部，17个附属的研究所和医院等。其间进行的个性化医疗和基因研究等项目往往需要海量的数据处理能力，不同的研究小组对IT环境和资源的要求也不尽相同。随着学院规模的扩大，IT部门面临着巨大的压力。就此，哈佛

医学院希望通过云计算解决在各个研究小组和机构之间实时、动态、按需调配计算资源的问题，以减少日常的管理和维护成本，将主要精力投入到与医学研究相关的任务上来。哈佛医学院将自己的IT平台搭建在Amazon EC2之上，形成了自己私有的“健康云”，在所管辖的不同研究机构之间实现共享。除此之外，该机构采用一系列成熟的云环境管理工具，将研究人员从底层的管理实施细节中解放出来，使管理成本降低到原来的20%左右。

2.7 云计算在制造行业的应用

经济全球化促成了全球化分工，一个产品的设计、研发和制造在不同国家进行。在改革开放初期，得益于低廉的人力成本和开放优惠的政策，中国迅速成为全球各大企业的代工中心。然而，随着中国经济的发展，沿海地区的人力成本优势正在逐渐消失，代工企业的利润也正在逐步降低，产业转型势在必行，需要寻找新的利润增长点。在产业转型的十字路口，许多厂商要么提高自身的竞争优势开始做自主品牌，要么将工厂逐渐转向其他人力资源成本更低的新兴发展中国家。鉴于竞争日渐白热化，厂商需要缩短产品的研发周期，让其在尽可能短的时间内上市。这个新需求将引入一系列的新问题，比如，如何协调供应链，如何互通文档数据等。

云计算为以上问题提供了答案。通过打造企业的私有云平台，制造业公司分布在世界各地的各相关部门能够将数据上传到云中进行共享和同步，通过与各个物料和零件供应商之间的公有云平台，随时了解它们的库存与市场行情，调整组装和备料方案，在最短的时间内完成产品的设计、生产和上市流程。除此之外，云计算平台还能整合企业内部的行政系统，电子化工作流程。世界各地的员工都可以通过私有云连接在一起，利用网络系统开会，减少差旅成本，进行协同设计。

传统制造业中的每个部门都各自独立，随着企业成长，对IT系统的依赖性逐渐增加，相应的IT系统也日趋复杂。用户往往需要面对不同部门的系统和流程，复杂性影响了信息的传递和企业对市场的反应速度。通过建立云平台，企业能够整合散落在各部门间的系统，使原本的系统虚拟化、集中化、自动化，并将其中的流程加以合并和连接，加速商业信息的流通，降低系统的建设费用。通过在云平台上的信息共享，企业能够与客户建立更紧密的伙伴关系，不仅让信息交流透明、通畅，更可以在第一时间掌握相关的市场动向和

客户需求，提高企业运营的效率。

广义的制造业还包含高新科技制造业，这些企业为了降低成本，纷纷寻求人力低廉的代工区域，形成设计和生产的全球布局，企业也从单一型走向全球整合型。即使是在代工工厂，也需要相应的IT系统来完成自动化生产、销售、采购、库存和物流等功能。在预算紧缩的情况下，高科技制造业开始选择云计算平台。通过对IT基础设施的共享，企业和代工工厂能够通过网络连接在一起。除了降低成本之外，这种模式还能协助企业在产能高峰和低谷时期的资源分配。云计算平台本身的分布式结构和按需使用资源的特性，使分散在世界各地的工厂对业务进行弹性的扩充或精简成为可能。

此外，云计算还能有效地保护知识产权。当企业向世界各地，尤其是向发展中国家进行迁移的时候，如果重新在当地建设新系统，不仅将耗费可观的费用，也不利于企业对专利技术的保护。通过建立自己的云计算平台，企业能够对资产和资源进行集中式的控制和管理，降低企业在专利技术上的风险。桌面云的部署方案如图2.6所示，这是我国台湾地区某制造业领军企业部署桌面云有效保护知识产权的具体方案。在该制造业公司实施云计算方案前，知识产权失窃的情况屡有发生，而实施此云计算方案后，再没有发生知识产权通过桌面系统（包括员工电脑等）失窃的例子，直接挽回的经济损失超过数百万美元。

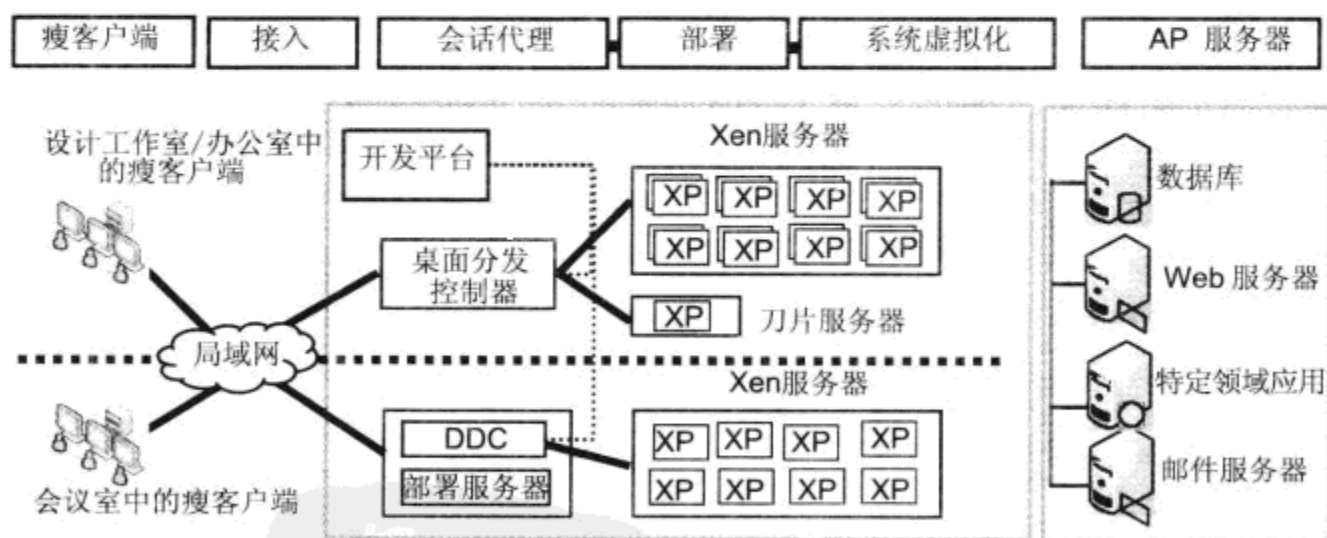


图2.6 桌面云的部署方案

中国互联网信息中心（CNNIC）在2010年7月份的《第26次中国互联网络发展状况统计报告》宣布，截止到2010年6月份中国网民达到4.2亿。而云计算作为一种通过网络提供的服务，正好契合了互联网公司业务的特点，因此云计算不仅成了互联网公司争相开展的新型业务，而且也对传统业务的进一步发展具有重要的战略意义。如今，互联网产业的

巨头们都对云计算倾注了极大的热情。互联网公司不仅仅通过业务创新提供形式多样的云计算服务，其本身也在使用云计算的平台和管理形式来运行其原有或者新型的服务。在众多的互联网公司里面，我们筛选出几个有代表性的来分析云计算给它们带来的机遇和价值。

首先，我们来分析一下Amazon。该公司是将基础设施资源作为公共服务的开创者之一，IaaS公有云服务能够被业界所认知，很大程度上归功于Amazon公司推出的AWS公有云服务。从如图2.7所示的AWS产品目录，我们可以看到AWS将复杂的应用解耦成各自独立且又互相利用的服务模块。这种解耦合的价值在于使得用户在各个服务模块上的调用和应用逻辑的编排变得简单，用户获得了较好的应用运行性能却无须了解系统实现的技术细节。



图2.7 AWS产品目录

AWS的EC2负责提供应用所需的CPU和内存资源，S3负责应用所需的存储，SQS负责应用模块之间的通信。有了这三者，应用开发就再也无须考虑应用所依赖的底层基础架构的技术特点。同时，该服务中的Elastic MapReduce和Auto Scaling技术能够帮助应用的开发者解决大规模并行处理应用与CPU相关的性能问题；EBS和Import/Export进一步解决存储时非关系型数据处理问题；而SimpleDB和RDS进一步解决存储时关系型数据处理问题；SNS增强应用模块之间通信的处理能力。在解耦应用后，AWS还提供应用公共服务模块，包括CloudWatch的应用监控、FPS的记账。按照这种思路，如果AWS下一步提供应用的安全模块、身份认证模块等，也将是水到渠成的结果。

如果综合使用AWS的整套服务，我们可以发现应用程序开发者只需要做两件事，一是业务逻辑的实现，二是业务逻辑的展现。应用开发将变得极其简单且只需专注于业务本身。我们通过一个实际的例子说明AWS云计算解决方案的具体应用和巨大优势。

作为互联网创业公司的代表，SnappyFingers.com的产品SnappyFinger定位为一个能够对互联网上的问答内容进行搜索的智能引擎，基本结构分为与用户进行交互的Web前端和支持用户查询的IRS（Information Retrieval System）后端两部分。Web前端由数个运行的EC2实例支持，与CloudFront一起提供Web服务器、数据库及页面缓存等一系列功能。网站后端的IRS则比较复杂，包含了用来获取Internet上问答数据的网络爬虫、相关的解析器及索引创建器等。为了满足大容量数据的存储和分析需要，IRS采用了Amazon的简单存储服务S3和简单数据库服务SimpleDB。SnappyFinger被设计成模块化系统，并采用了Amazon的简单队列服务SQS，用于在不同功能部件之间进行通信。

通过使用AWS，网站开发人员能够将主要精力集中在搜索算法上，无须去考虑诸如服务器部署、数据存储或消息可靠传递等一系列底层问题。传统的重量级互联网应用在AWS的支持下，变成了小团队就能够开发和运营的业务。

SnappyFingers作为一个新创公司使用云计算提供的资源可以快速构建其自身的IT系统并且推出自己的业务。而大型的互联网公司，也需要通过云计算系统来实现资源共享，提高效率并降低成本。

Google公司是互联网公司中提供Web应用及其应用平台的代表。Google提供了大量的Web应用，其业务范围涵盖了传统的搜索业务、在线电子邮件业务、在线文档、在线地图等等，不仅如此，Google还开放了应用平台Google App Engine，该平台能够提高Web应用的可扩展性和可用性，大大方便了普通的Web应用开发人员。而这些云计算服务都依赖于Google自身基础设施架构和应用服务的设计对可伸缩性和高可用性的充分考虑，从而达到了效率与成本的平衡与优化。例如，海量的数据存储在与数量众多的节点上且通过这些节点的合作来完成搜索任务。这些节点运行在Google特别设计的硬件服务器上，这些服务器的成本比普通服务器低很多（约为普通服务器的1/5到1/20）。而整个系统可以容纳任意节点的故障，一台服务器坏掉了直接拔出来更换掉即可。Google的基础云架构实现了业务集成和资源共享，在支撑超大规模应用的同时减少了运行管理成本。

2.8 云计算在教育科研领域的应用

教育科研是一个国家保持可持续发展和创新的基础，也是全社会关注的重点。教育科研领域的信息化建设就是要采纳最新的信息技术，实现广泛的合作，促进先进的教育科研成果的流通，从而提高教育效果，加快科技进步。

一般来说，教育可以分为三个阶段：以保证人口素质为目标的普及性基础教育，以培养人才为目标的高等教育，以不断更新和提高为目标的继续教育。在这三个阶段中，教育的目标和方式各不相同，对信息化技术也提出了不同的要求。但不论哪个阶段，提供教育服务的大学、中小学、职成教等机构的主要工作内容包括课堂教学、实验和教辅三大领域。

2.8.1 云计算在课堂教学领域的应用

传统的课堂讲授方式中，老师通过口述并运用板书配合讲解，学生缺乏对教学内容的真实感受。为了改善教学效果，还需要通过多种方式来增强学生对于教学内容的亲身感受，提高动手能力。近年来，利用各种多媒体方式来展示教学内容已较为普遍，通过这种方式，可以大大提高教学内容的直观性，增加教学的互动性，激发学生的兴趣、想象力和创造力。然而，共享这些丰富的教学内容需要高效、普遍的信息化基础设施，云计算能够为教育的信息化建设提供技术支撑和交付方式。

从教育资源的分布状况来看，若采用大范围分散式的模式，投入巨大且效率不高。因此教育行业可以采用集中式的信息化基础设施，通过网络远程访问，实现优质教学资源的共享和新型教学方式的推广。这种方案既可以提高投入的效率，又可以促进资源的公平分布，特别是提高边远和落后地区的教学效果。通过云计算搭建“教育云”平台，是教育信息化建设的重要方向。在我国台湾地区试行的“电子书包”计划就是云计算在教学领域的典型体现。为了减轻小学生背书包上学的负担，台湾地区的教育部门自2009年8月1日起开始推行“电子书包实验教学试办学校暨辅导计划”，在未来2年内选定5所小学进行试点。学生们将得到一台轻薄、具有触控式屏幕的电脑，通过网络与老师在校园内随时随地进行移动式学习。网络连接的后端即为教育云平台，平台内存储着大量的教学资源，如与课本内容同步的资料库，以及师生间互动的空间和工具等。

2.8.2 云计算在教学实验中的应用

实验是教学中的重要一环，学生通过动手实验来获取知识，探索新的领域。然而，学校拥有的资源常常不能保证每个学生都拥有自己的实验室。而云计算通过共享开发测试资源和远程桌面共享的方式，可以很好地实现“每个师生拥有一个虚拟实验室”的梦想。基于云计算的“虚拟实验室”工作原理如图2.8所示。从图2.8中可以看到，虚拟实验室通过标准化环境建设完成实验室环境准备，通过虚拟化资源池建设完成实验室环境搭建，通过自动化方式完成实验资源申请、回收、监控和管理，通过虚拟桌面的方式完成远程访问。

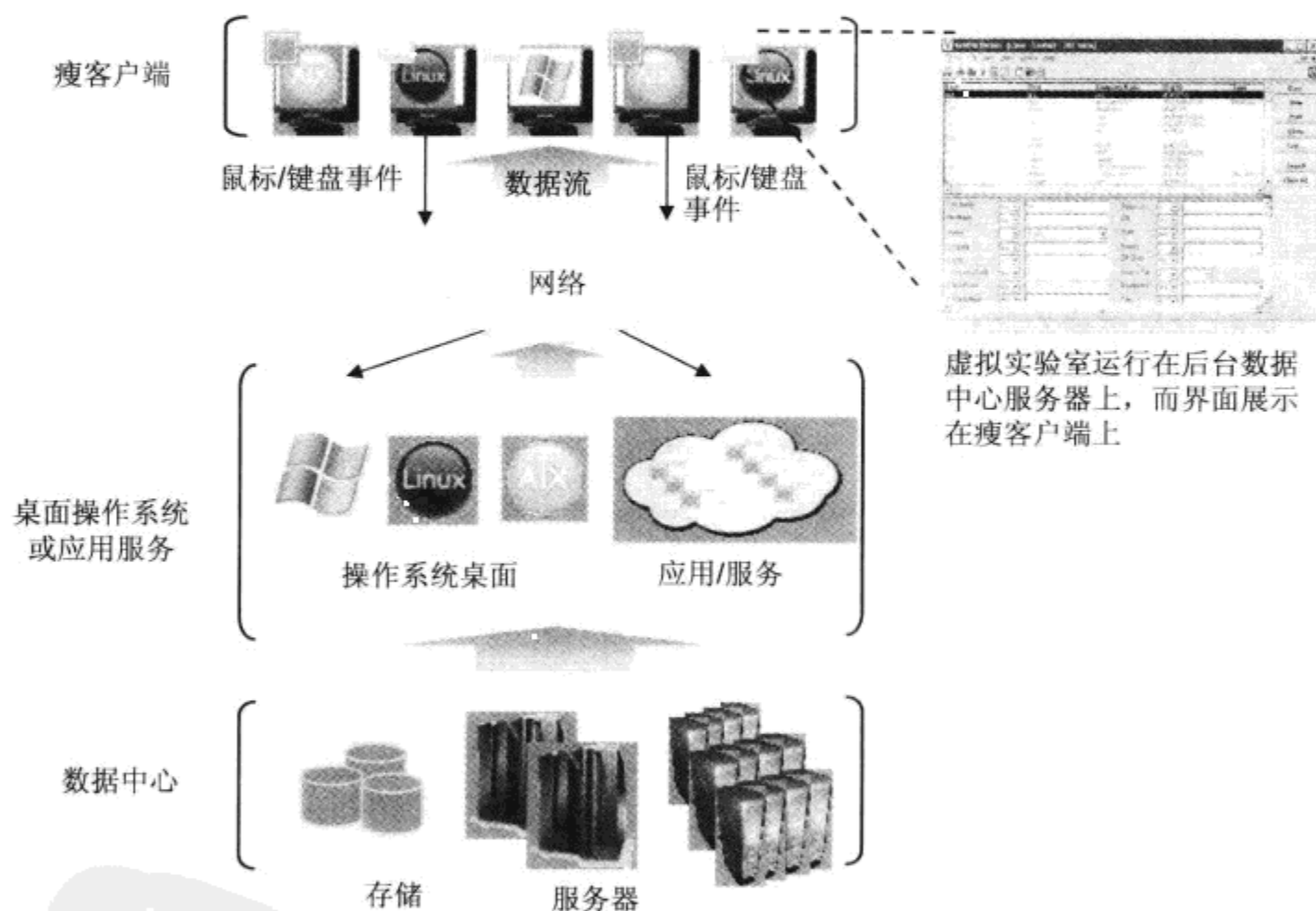


图2.8 基于云计算的“虚拟实验室”工作原理

实际上，这种虚拟实验室的方式在发达国家已经开始建立。比如，在2008年7月30日，美国国家科学基金会（NSF）的计算机信息科学与工程中心（CISE）宣布将资助伊利诺伊州立大学（UIUC）在位于巴那市和香槟市之间的校园内建立云计算实验中心。该平台将由伊利诺伊州立大学管理，作为开放的资源提供给其他从事数据密集型计算研究的机构使用，例如医学、生物学、物理学、气象学 and 经济学等。

2.8.3 云计算在教辅领域的应用

教育云能提高学校的行政管理能力，整合学校的各种信息化系统，如办公自动化系统、学生信息系统、教学管理和教育效果评估系统等。通过这样的整合，学校管理者可以有效地监控教学的效果，分析相关的原因、寻求改进提高的方案。

综上，我们可以将云计算在教研、教学和教辅三大领域的应用统称为“教育云”。打造教育云对于一个国家来说是重要而迫切的，因为在培养国家认同感和国家竞争力方面，教育向来是重要的基石。教育云所蕴含的资源 and 能量，不应只局限在一个国家的内部，而应从全球的角度来考察。教育云除了能让学生方便地接触到全世界的知识，还能使教育机构进行跨国界的交流与合作，进一步强化教育内容体系，对内提升国民对传统文化和国家的认同感，对外实现交流文化、科技乃至价值观的目标。

要打造一个成功的“教育云”，除了政策的扶持、足够的经费和相关的技术之外，还有如下一些关键要素：

- ▶ 使用者的教育方式和观念。相比于传统的教育模式，老师需要更多的创新意识，学生需要更多的互动，家长也需要通过此平台与学校、老师沟通，进一步了解孩子的学习情况；
- ▶ 政府的积极介入和管理。政府需要扮演引导者和推广者的角色，比如，建立能够提供符合教育云要求的教师培训体系；提供相应的政策对先行者进行补助或奖励，从而鼓励更多企业和学校参与其中；
- ▶ 健康的产业链及生态环境。独立软件开发商（ISV）、IT硬件设备生产商、数字内容提供商需要共同参与教育云的建设和维护，从而创造出优质的“教育市集”，这样不但可降低整体成本，还能够培育出优秀的教育服务；
- ▶ 适应现代信息化环境的教育教学方式。教师要有意识地利用信息化的教学手段，对学生因材施教，教育产业应该开发丰富多样的信息化教学内容，学校、社会和家庭应该利用信息化手段实现经常性的互动。

2.8.4 云计算在促进科研合作中的应用

科研既是高度个人化的活动，又需要广泛的合作。大型科研机构一直注重促进科研人

员之间的合作，而合作平台的建设是其中重要的一环。云计算的出现，实现了全球范围的资源和能力的共享，也成为科研合作的一个重要平台。下面以IBM全球研究机构所设计和建设的云计算平台为例来介绍云计算在促进科研合作中的作用。

IBM研究部门分布在全球9个国家，各自拥有一定的IT资源。在实现对IT资源的有效管理以满足不断增长的需求时，研究部门的IT管理面临以下的需求。

(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；

(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；

(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：应该简化管理复杂度，需要在任意地点通过网络进行有效的远程管理。

为了满足大规模的计算任务对资源的需求，IBM研究部门集成分布在全球的九大研究机构（如图2.9所示）中的IT资源形成一个基于Web的私有云：IBM Research Compute Cloud（RC2）。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。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按需应变、自助获取的计算资源，成为云计算创新的试验床。



图2.9 IBM 全球研究部门

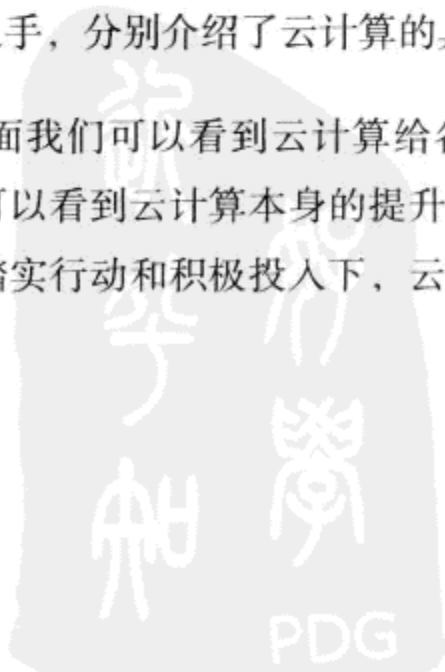
RC2提供的核心功能包括：服务的设计、注册和管理，服务目录注册和访问管理，大规模的分布式程序模型和开发环境，镜像管理，计算、存储和网络的集成虚拟化，资源的动态管理，大规模的计算能力，以及大规模的存储。RC2提供了解决方案按需应变的自助交付，支持对服务整个生命周期的管理，包括服务的创建和订阅、资源的动态整合、服务监控和计费等，它基于业务流程机制灵活处理用户请求。RC2可以简化基于Web的编程模式，优化应用程序设计，使得基于Web的应用和服务的创建、交付和管理变得更为简单，隐藏掉IT管理的复杂度，简化资源的分配和使用，提供灵活可扩展的大规模IT组件构建的基础架构，根据负载特征优化计算环境。

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、虚拟化资源优化整合等）。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，从而实现优化的云中存储。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，从而实现资源和服务以透明和动态的方式被管理、部署和重新分配。

2.9 小结

本章首先回顾了云计算的技术本质和业务本质；接着从宏观层面描述了企业在实施云计算时应关注业务、技术和风险等三大要素；然后从政府、银行、电信、物流、医疗、制造、互联网、教育等八大行业入手，分别介绍了云计算的具体实践和最新进展。

在分析这些现状时，一方面我们可以看到云计算给各行各业正在带来和业已带来的深刻变化；另一方面，我们也可以看到云计算本身的提升空间。我们可以大胆地预测，在各行各业从业者的深入思考、踏实行动和积极投入下，云计算在未来几年将迎来更高的高峰，创造出更大的价值。



第 3 章

云计算的实施

- 3.1 云计算的实施要点
- 3.2 企业实施云计算的战略规划
- 3.3 企业云计算业务的实施
- 3.4 云计算提供商的业务模型
- 3.5 云计算提供商的平台构建
- 3.6 云计算平台的运维管理
- 3.7 小结

PDG

云计算体现了以服务方式变革IT的新思维，一系列业界具体实践已在证明这种新思维的价值。本书除了让读者认识和思考云计算外，也希望在读者进行云计算实践时给予一定的帮助和指导。因此，本章尝试分析在云计算实施过程中所应遵循的一般方法和特别之处。3.1节首先介绍云计算提供商和企业用户在实施云计算过程中所应关注的要点。这些要点分别从云计算提供商和云计算用户两个角度，以及战略、控制和执行三个层面进行阐述。然后，本章就实施要点的关键之点展开分析，分别从企业用户和云计算提供商的角度来看实施云计算所应关注的几个重要方面。需要指出的是，与通常的IT系统和服务的实施相比，本章重点强调因云计算的特点而引起的不同之处，对于具有共性的一般问题，读者可以参考其他相关资料。

云计算的实施涉及资源池、云管理平台、管理员、用户和建设者这五大要素，如图3.1所示。下面简要介绍和分析这五大要素。

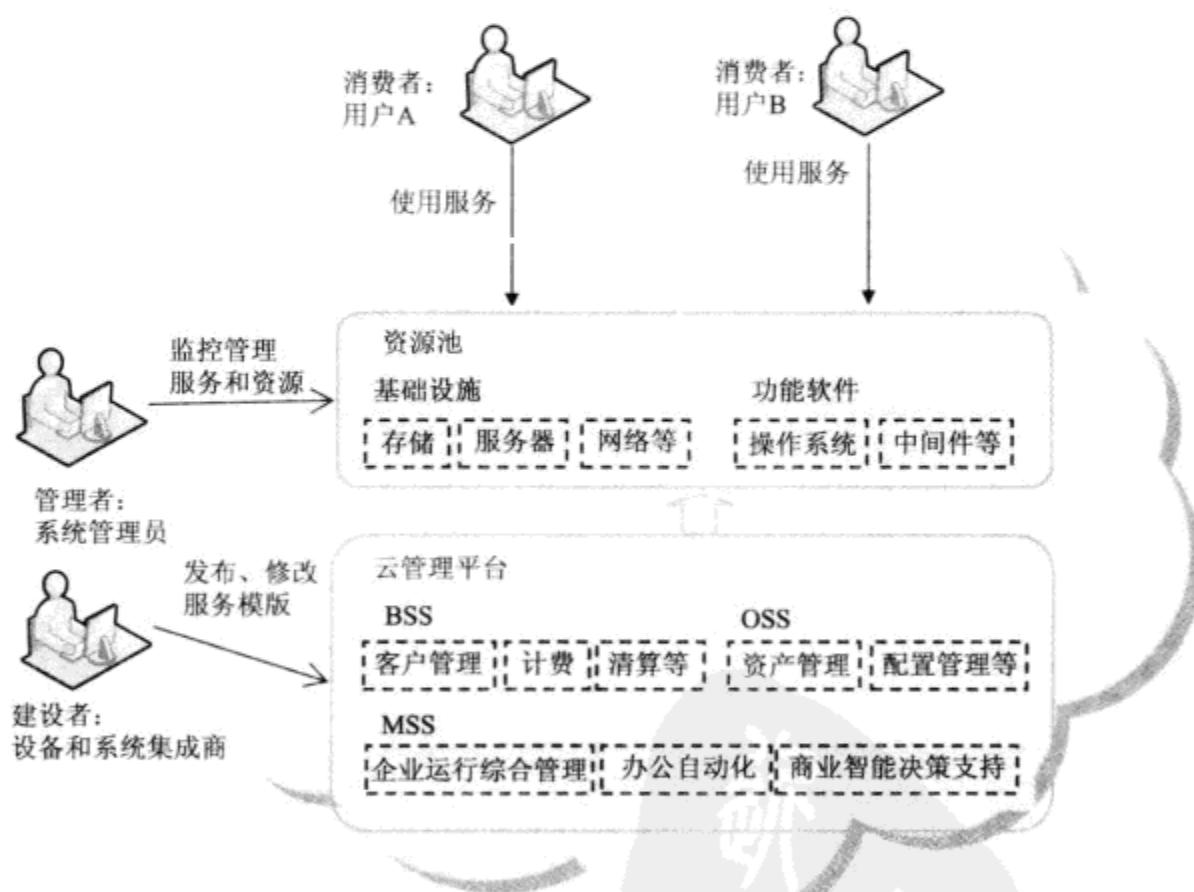


图3.1 云计算实施的组成要素

1. 资源池

指数据中心的基础设施和功能软件，包括服务器、存储、网络、安全设备、操作系统、中间件等。在建设云计算系统时，资源池的构建受多种因素影响而差别很大，包括计

算资源数量（如上万台服务器还是几百台服务器）、用户群规模（如几百万用户还是几万用户）、所提供的功能（如IaaS、PaaS还是SaaS）等。管理资源池时通常采用虚拟化方式提供统一的管理接口，屏蔽来自不同厂商产品的差异。

2. 云管理平台

是云计算服务提供商提供和管理云计算业务、云计算消费者订阅和配置所需服务的平台。一般情况下，云计算管理平台分成BSS（Business Support Service，业务支撑服务）、OSS（Operation Support Service，运维支撑服务）和MSS（Management Support Service，管理支撑服务），关于云计算平台的介绍将在3.6.3节中详述。

3. 管理员

主要是进行云计算管理的运维人员及其相关的IT经理。管理员监控和管理云计算提供服务的使用情况并对IT设备进行日常运维。在云计算业务的管理中，管理员管理的对象本质上是服务而非设备，而服务又直接支持业务。因此，管理员具备业务能力非常重要。

4. 用户

不同的云计算业务对应的用户类型不同。比如，IaaS的用户是自动申请资源的中间件或手工调用资源的IT工程师；PaaS的用户是自动部署应用的上层软件或手工部署应用的IT工程师；开发测试云的用户是软件开发测试团队；桌面云的用户是企业的普通员工。针对不同的用户，云计算所应提供的虚拟化程度、自动化流程和性能指标也应不同。在实施云计算时，分析确定用户特点非常重要。

5. 建设者

一般是系统集成商或服务供应商。建设者根据需求设计云计算服务，并进行具体的系统集成和设备调试。企业为了适应不断变化的环境和业务，其IT能力需要不断地进化。在云计算时代，企业IT能力的进化表现在云计算业务的快速、持续更新。系统集成商作为云计算的建设者，其快速、持续的建设能力，包括技术能力、集成商自身的财务能力、持续发展能力等，在云计算项目中特别重要。

另外有一点需要特别指出，从本质上，云计算实际上也是某种类型的IT系统和服务，它在实施的过程中也需要遵循IT系统开发、测试、运行、维护所应遵循的一般方法。

3.1 云计算的实施要点

在当前的IT实践中，系统的功能变得越来越强大，结构也变得越来越复杂。为了顺利建设和使用IT系统和业务，应该遵循一定的方法和原则。我们在本节分析实施云计算应该关注的要点。

云计算实施中应关注的要点如表3.1所示，它从两个角度、三个层面来说明云计算实施的要点。两个角度是云计算提供商和企业用户的角度，三个层面则来自战略、控制和执行。在战略层面，云计算的提供商和企业用户应该根据企业的自身条件和已有资源，结合企业的业务特点和发展方向制定实施云计算的指导思想和组织架构。在控制层面，为了实施云计算，企业用户应该制定相应的规划、规章和制度规范云计算实施过程，控制云计算实施的方向和进度。在执行层面，云计算提供商或者企业用户应关注云计算实施的执行过程，即在云计算系统和业务的设计、开发、建设、运行过程中的具体步骤和操作。在执行过程中需要遵循所制定的控制措施，而控制措施的制定也需结合云计算在执行过程的特点。

表3.1 云计算实施中应关注的要点

		战略（指导思想）	控制（规划、规章和制度）	执行（实施和监控）
云计算企业用户	采用云计算	价值分析，风险分析	业务适应性分析，制定实施步骤	实施应用层云计算业务，实施平台层云计算业务，实施基础设施层云计算业务
云计算提供商	业务战略（BS）	客户需求识别与业务发展战略（业务模型），管理架构	需求规划，营销规划，容量规划，绩效规划，业务管理流程和控制	客户服务体系的建立和云计算服务销售，业务绩效考核与分析
	业务支撑（BSS）	服务与方案实施战略，客户服务战略，营销模式	服务流程控制，客户关系管理制度，营销策略，定价策略	客户管理，产品管理，定价管理，订单管理，合同管理，订阅管理，计费管理
	运营支撑（OSS）	服务交付战略，服务保障战略，基础架构资源战略	基础架构资源规划，服务等级协定（SLA）	服务模板管理，虚拟镜像管理，服务自动化管理，配置管理，事件管理，变更管理，问题管理，SLA管理，计算、存储和网络资源管理
	管理支撑（MSS）	信息安全与审计的战略和组织架构企业管理架构	安全事件和审计控制，隐私和数据保护规则，企业管理的流程控制	用户身份与访问控制，入侵检测，安全审计执行，企业综合管理，办公自动化和知识管理

在云计算实施中，云计算提供商将提供云计算服务作为其核心业务，在构建提供云计算服务的IT系统时需要考虑以下四个方面的问题。

- ▶ **业务战略**：分析规划云计算业务，制定业务的发展方向和盈利模式，搭建组织架构以具体运行云计算业务。
- ▶ **业务支撑**：对服务产品和客户的管理，通过一定渠道让用户了解、购买并使用相关的云计算服务。
- ▶ **运营支撑**：关注云计算服务的上线开通，保障服务的正常运行。
- ▶ **管理支撑**：关注企业的运行流程和日常运转所需的知识和系统支持。

从云计算企业用户的角度来说，他们使用云计算的过程将变得简单，这也是云计算的本质优势之一。云计算用户需要从战略层面考虑采用云计算的价值和风险；在确定采用云计算方案后，需要从控制层面考虑所需业务对云计算的适应性，并制定相应的实施步骤；在具体的执行过程中可以根据不同层次云计算业务的特点，充分利用云计算平台所提供的工具和方法来构建、部署和运行云计算业务。

由于篇幅所限，本章将不对表3.1中的所有要点逐一展开分析，仅选择从企业用户和云计算提供商两个角度来看实施云计算中的几个关键点。对于企业用户来说，3.2节介绍了实施云计算的战略规划，涵盖了表3.1中战略和控制两个层面的内容；3.3节从执行层面展开来分析企业用户实施云计算的方法和步骤。对于云计算提供商来说，业务战略的关键是设计自己的业务模型，而业务实施的关键是构建和运维管理云计算平台，因而3.4节介绍了云计算提供商的业务模型设计；3.5节介绍了构建云计算平台需要重点关注的方面；3.6节详细介绍了运维管理云计算服务的平台和方法。我们希望本章内容对相关读者在开展云计算业务时有一定的参考价值。

3.2 企业实施云计算的战略规划

IT系统是企业运营的重要支撑。由于云计算模式具有鲜明的特点和优势，当前大量的企业都在考虑采用或者过渡到云计算。企业在向云计算过渡时需要制定明确的战略目标和

有效的实施规划。事实上，制定企业的云计算实施战略及规划就是在制定企业的IT基础设施架构和信息系统发展蓝图，对企业业务的发展具有长远影响。制定好企业的云计算实施战略及规划并不是一件容易的事，对于具有一定规模的企业而言，IT系统向云计算转型是一项复杂的工作，往往需要依靠在云计算领域有丰富实践经验的技术专家和咨询人员来完成。本节向读者展示一个指导企业制定云计算实施战略规划的系统化方法，旨在让读者了解云计算战略规划的一般方法和原则，并不探讨评估和分析的技术细节。

3.2.1 战略规划概述

实施云计算并不是对数据中心技术的简单革新，而是对IT运用模式的根本性改变。云计算模式需要企业将自己对IT资源和数据的控制能力转移到云计算平台。企业决策者需要考虑实施云计算的价值、风险、工作量、业务适应性、对企业组织结构的影响等多方面因素。设计企业的云计算战略需要系统化的方法。无论是自己制定还是雇佣咨询公司，企业自身了解该方法都是必要的。以下列出了云计算实施前战略规划中的几个关键步骤及所能解决的问题。

- ▶ 云计算实施的价值分析：企业通过云计算实施的价值分析，能够清楚地判断是否需要实施云计算，并确立实施云计算应该达到的业务目标。
- ▶ 云计算实施的风险分析：企业通过云计算实施的风险分析，能够清楚地了解企业实施云计算可能会引发的问题，明确不同业务系统在迁移到云计算平台的过程中需要面临的各种关键问题，做到未雨绸缪。
- ▶ 云计算实施的战略定位分析：云计算有不同的层次，也有不同的类型。不同的企业实施云计算有不同的目的，在云计算生态系统中扮演着不同的角色。云计算战略定位分析将帮助企业明确自身在云计算生态系统中的位置和角色。
- ▶ 云计算实施的适应性分析：一旦企业确定了云平台类型和自身的定位，下一步就要考虑如何将现有的IT系统迁移到云平台上。云计算业务适应性分析可以帮助企业分析出迁移过程的优先级。

企业实施云计算不可能一蹴而就，是一个长期渐进的过程。企业自身有着不同类型的

应用系统，需要将候选的业务系统排出优先级，先后分步进行。企业可以根据云计算实施价值分析、风险评估及业务适应性分析的结果，选出可获取较大价值，同时风险较小并易于实施的业务系统，优先实施云计算。在具体实施的时候需要按照规划，制定相应的计划和路线图。

3.2.2 价值分析

云计算实施的真正意义在于为企业的业务和运行带来价值，例如为企业节省IT开销、提高业务系统的能力等。企业在考虑采纳云计算时，首先需要进行云计算实施的价值分析。一个云计算实施和价值分析工具如表3.2所示，它列出了价值分析的维度和相关的评估点。每一个维度是一个问题域，每一个问题域包括了若干个问题。当然，表3.2所列为通用的情形，具体到某个企业还可以根据企业的行业特色和竞争力做相应的修改。决策者通过回答这些问题，评估实施云计算是否能给企业带来真正的价值。

表3.2 云计算实施的价值分析工具

问题域	问题：“企业是否需要……？”
组织与人员	精简现有组织，减少IT管理人员和相应的人力成本 精简现有组织，减少IT开发测试人员和相应的人力成本
IT服务质量	提高IT服务的性能 提高IT服务的可用性 提高IT服务的可扩展性
业务敏捷性	加快应用的开发、测试 简化应用部署 实现业务快速上线 能否减少业务升级更新的时间
IT开支	降低数据中心构建开销 通过减少服务器购置节省开销 通过减少软件购置节省开销 降低IT系统安装配置的开销 降低IT系统管理的复杂度

实际上，表3.2的问题体现了云计算模式可能给企业带来的价值，企业需要判断自身是否需要这些价值。云计算倡导企业和组织以租用服务而非拥有系统的方式来实现企业的信息化，以降低企业IT服务的开销。企业使用云计算服务能够获得的价值主要体现在以下几方面。

- ▶ 精简IT部门组织架构和人员，降低IT系统复杂性。企业将IT系统迁移到云计算平台上，可以由云计算平台承担IT系统大量的运维工作，如日常维护、管理数据库、升级更新、为系统和软件打补丁等。这样，企业的IT团队就可以将主要精力放在核心业务的创新上，专注于提高企业的核心竞争力。云计算将IT系统的复杂性转移到云上，将简洁易用留给用户。
- ▶ 享受更高质量的IT服务。云计算为用户提供更统一、更专业的服务。例如，自动化管理技术可大幅度提升管理效率；自助服务方式使用户通过登录云平台并随时提交服务请求，即时获得所需的云计算服务；统一的安全管理、按需响应的动态伸缩保证服务性能的一致性。
- ▶ 业务敏捷性大幅提升。云计算采用的虚拟化、标准化和自动化技术将使企业获得IT服务的灵活性大为增加。例如，通过虚拟器件、快速部署、动态伸缩等一系列技术，用户可实现业务系统的快速上线、快速配置、快速升级，并可适应实时动态变化的负载。标准化的接口使业务系统之间可以更容易地集成，实现服务的快速组合和构建。
- ▶ 降低IT系统投资。企业在使用云计算提供商的服务时，无须先期投资建设IT基础设施，仅需订购相应的服务即可使用。这样可以大幅度缩减构建数据中心、购买服务器、网络设备、软件等一次性IT投入。此外，传统的业务应用大多按照应用的峰值负载来设计，造成平均利用率较低。通过采用云计算来满足高负载对计算能力的需要，企业可以减少自身IT系统的容量，降低IT投资的浪费。根据IBM研究部的一项报告，对大多数中小企业来说将测试工作转移到私有测试云上，可节省65%的硬件投资，27%的软件投资，45%的系统管理开销和多达76%的应用部署开销。

3.2.3 风险评估

在明确了实施云计算可以带来的价值后，企业还需评估实施云计算可能带来的风险。表3.3是一个云计算实施的风险分析工具，它列出了风险评估的维度和相应的问题。每一个维度是一个问题域，每个问题域包含若干问题。表3.3所列举的是典型的问题，使用时可以根据具体需要增加相应的问题域和问题。决策者通过回答这些问题，评估实施云计算可能

面临的风险，并找出可能的应对方案。

表3.3 云计算实施风险分析工具

问题域	问题：“通过实施云计算，是否存在……？”
IT服务	云计算提供商满足适当的服务等级协定的风险 云计算提供商保证云服务连续性的风险
数据	敏感数据在云计算平台的安全的风险 云计算平台的安全控制措施可靠可行的风险 企业IT系统和云计算平台之间数据迁移安全方便的风险（例如使用某些SaaS服务时，应用产生的数据是否可以导出） 数据丢失的风险
成本	过渡到云计算服务所需的系统升级和技能培成本过大的风险 企业现有应用迁移到云计算平台的成本过大的风险 企业现有应用与云计算服务集成成本过大的风险
政策法规	将业务系统迁移到云计算平台违反公司内部政策的风险 将IT设施和数据置于自身控制之外违反国家法律法规的风险 云计算服务难于控制和审计的风险

对采纳云计算的顾虑很多都来自于对业务连续性和数据安全性的担忧。当企业将IT系统迁移到自身控制范围之外时，它也就失去了很大一部分对业务和数据的掌控能力。例如，由于企业通过网络访问云计算服务，而云计算服务运行在远端，云计算服务的可靠性，网络通道的可靠性都将影响到业务的连续性。此外，由于企业无法知道数据存储在哪些物理资源上，也不知道其所在的数据中心和地区，伴随而来的顾虑是数据会不会被人随便访问，会不会丢失，能否随时导出等。企业进行风险评估时，需要重点了解云计算提供商在这方面的措施和能力。

对于使用者至关重要的特性在不同的云计算提供商和平台之间存在着很大差异。成熟的云计算平台可以在多个地方分别创建多达3个以上的数据备份，进行实时恢复，并可定制数据存储的约束条件。用户还可以通过标准的数据同步接口，将云上的数据备份到私有的数据中心的存档。云计算提供商建设在不同位置的数据中心有不同的网络接入，从而尽量减少某些网络连接中断对业务连续性的影响。总之，企业应该充分关注云计算服务可能带来的风险，并且考虑到相应的应对之策。这些应对的措施一方面是云计算提供商所应具备的，一方面也是企业自身应该准备好的。

3.2.4 战略定位分析

如果说云计算实施的价值分析和风险分析是帮助企业理解为什么要实施云计算，那么战略定位分析则是要帮助企业判断实施云计算是否是可行的战略选择。这个判断主要包括三个方面：

- (1) 企业选择的云计算服务的部署模式，即云计算的服务方式；
- (2) 企业选择云计算服务的层次，即云计算的服务类型；
- (3) 企业在云计算服务中承担的角色，即是云计算的提供者，或是使用者，或是两者兼具。

对于云计算服务的部署模式，如在本书第1章中所介绍的，可以根据云计算平台跨越组织边界的程度分为私有云、公有云和两者兼具的混合云。私有云和公有云的一个直接区别就是区分云服务的使用者和提供者是否存在于同一个企业之中。

对于云计算服务的架构层次，如本书在第1章中所介绍的，由高到低依次为：应用云提供的软件即服务（Software as a Service, SaaS）、平台云提供的平台即服务（Platform Cloud Services, PaaS），以及基础设施云提供的基础设施即服务（Infrastructure Cloud Services, IaaS）。

对于企业在云计算服务中承担的角色，它可以是云计算服务的使用者，采用云计算的目的就是使用云计算服务来实现企业自身的运行需求。另一方面，企业也可以承担一部分或者全部的云计算提供者角色，如私有云就是由企业自身建设和维护的。企业所开发的云计算服务部署到云计算平台后也可以开放给第三方用户。

云计算服务的部署模式，云计算服务的架构层次，以及企业在云计算服务中所扮演的角色，即为云计算分析中的三维模型。企业的战略定位分析是在这个三维模型上找到自己的定位和角色。该定位分析由许多因素决定，例如：企业的愿景和战略规划、企业的规模、企业在行业中的地位、企业的业务模式、企业IT基础设施的成熟度、企业组织上的成熟度等。比如，假设某级城市政府职能部门中的信息中心负责该市政府电子政务系统的建设与维护，该电子政务系统要为其他各委、办、局提供IT服务，因

此希望将电子政务系统迁移到云平台上，由信息中心来管理云平台中的应用。在这个例子中，该市的市政府所要构建的是一个跨部门的云计算环境，服务的提供者和使用者在同一组织内部，所以属于私有云中的企业云。其中信息中心是服务提供者，各委办局是服务使用者。

专业的云计算咨询公司可以为企业在云计算战略定位上提供帮助，但最终决策还是由企业自己做出的，毕竟这种定位将反映企业在未来所要支持的商业模式。企业根据价值分析和风险分析的结果来决定自己使用什么样的云计算服务部署模式，利用什么层次的云计算服务，承担什么样的使用者或者提供者角色。企业正确区分并选择自己在云计算服务中的战略，对企业成功地实施云计算并有效地服务于其核心商业目的是至关重要的。

3.2.5 业务适应性分析

在确定实施云计算后，企业应该分析现有IT系统和IT服务的类型与特征，从而确定哪些业务系统或IT服务适合云计算的实施，据此选择合适的云计算平台类型和服务提供商。在准备将某候选业务系统转换为云计算实现之前，企业需要对业务系统进行云计算适应性分析，即分析业务和业务系统的各种特性，然后做出综合判断。本节从应用负载的角度给出将应用迁移到云计算平台进行评估和规划的概念、过程和方法，称之为应用负载云策略及云转型分析。

图3.2给出了应用负载云策略以及云转型评估和规划分析的具体步骤和方法。首先，采用应用负载云策略组合分析，对企业中的应用负载进行全面系统的梳理，按照应用负载的分类制定云策略。分析后的结果包括哪些应用负载适合采用云计算服务，哪些则不适用；哪些应用负载采用云迁移策略，哪些采用云转型策略。所谓的云迁移即将应用从现有IT系统迁移到云计算平台上而无须显著改动，而云转型则指对现有应用进行显著改造后才能运行在云计算平台上。接下来，对于采用云转型策略的应用负载逐一进行重点分析，评估该应用负载进行云转型的基本方案及计划。这部分的分析结果包括选定的云服务提供商、云架构类型、必要的转型步骤和转型工作量等。

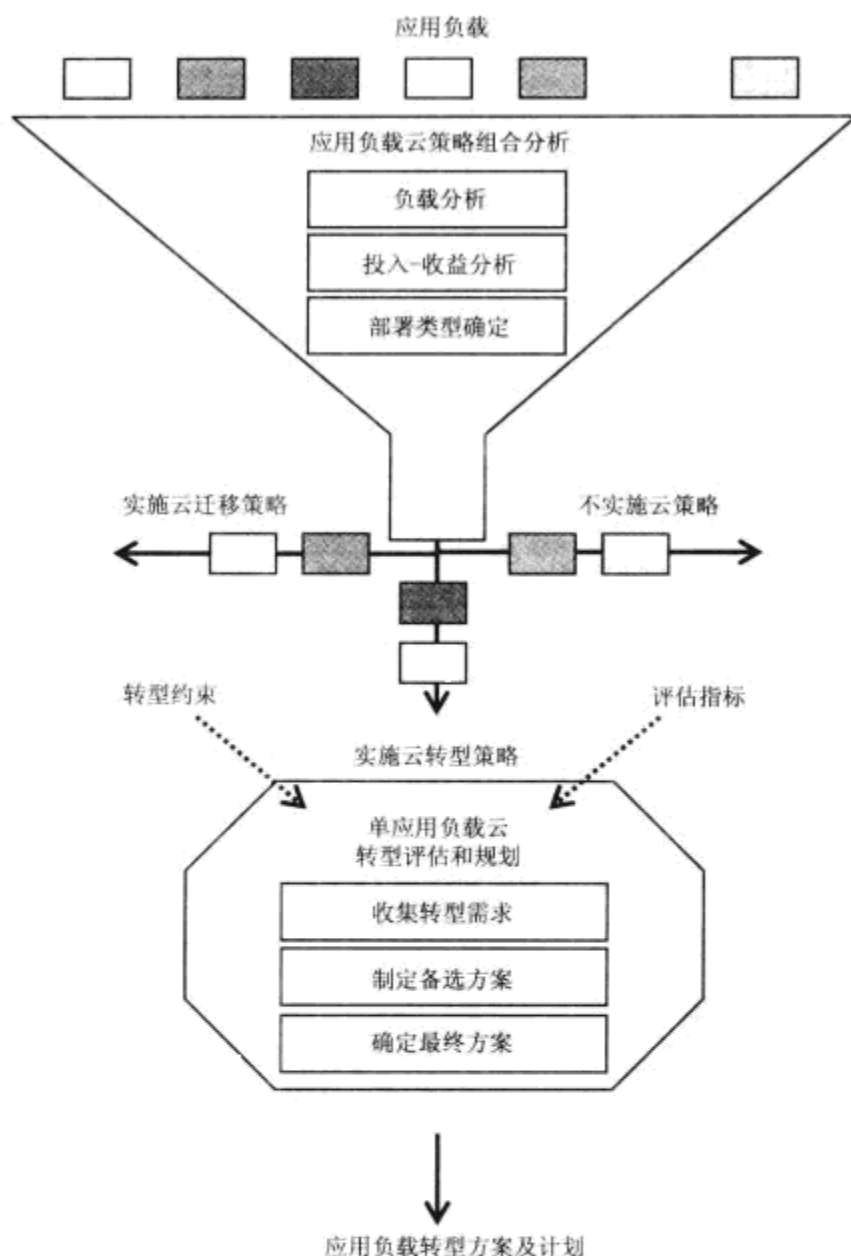


图3.2 应用负载云策略和云转型评估和规划过程

3.2.5.1 应用负载的云策略制定

制定应用负载的云策略主要是回答两个问题：第一，该类应用是否适合采用云计算服务；第二，该类应用采用云迁移策略还是云转型策略。这两个问题对于成功制定云计算的战略规划至关重要。本节着重介绍应用负载云策略组合分析的概念和方法，并采用该方法制定出应用负载的云策略。

应用负载云策略组合分析是指对企业内所用的应用负载进行分类和特性刻画，然后根据云计算的要求对应用负载进行组合，并对每一类应用组合进行“投入-收益”分析，最终确定各类应用负载所应采用的云策略。

事实上，并不是所有的应用负载都适合使用云计算服务，而且不同的应用负载可能会

采用不同的云服务交付模式。一份IBM市场洞察力分析报告对各类应用负载进行了分析，给出各类应用负载适合的云交付模式。通常，IT基础设施以及协同式的应用负载更适合部署在公有云上，而对企业更为重要的数据库、业务应用以及相应的基础设施更适合部署在私有云上。表3.4给出了一些典型的应用及所适合的云实施策略。

表3.4 不同应用负载云实施策略实例

适合公有云的应用负载	适合私有云的应用负载	不适合实施云的应用负载
<ul style="list-style-type: none"> · 音频/视频/网络会议 · 服务咨询 · 用于培训和演示的基础设施 · 广域网以及VoIP基础设施 · 桌面应用 · 用于测试的基础设施 · 存储 · 数据中心的网络 · 服务器 	<ul style="list-style-type: none"> · 数据挖掘、文本挖掘以及其他分析类应用 · 安全类应用 · 数据仓库或数据集市 · 业务连续性以及灾难恢复系统 · 测试环境的基础设施 · 长期数据归档和保存 · 交易类数据库 · 行业相关的应用 · ERP应用 	<ul style="list-style-type: none"> · 企业内高度敏感的数据以及管理和处理该类数据的应用 · 高性能的在线交易处理 · 法律法规需要审计的数据以及管理和处理该类数据的应用 · 未提供虚拟化或云上使用许可证的第三方软件

在了解了各类应用负载可能会采用不同的云策略后，需要对企业内的应用负载进行全面梳理。首先，根据应用负载的特性将应用负载进行分类组合。通常企业中的应用负载分为以下五大类，如图3.3所示。

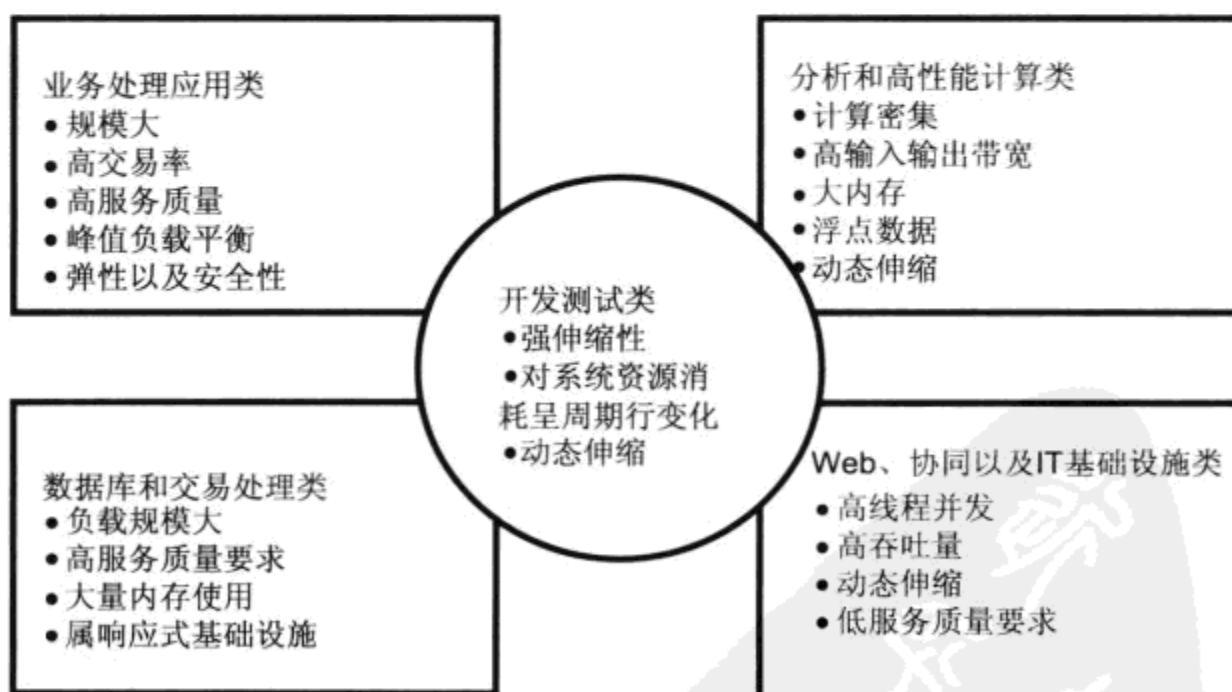


图3.3 典型的应用负载分类，开发测试类的应用来自于其他应用类型

- ▶ 业务处理应用类：该类应用负载通常支撑着企业主营业务活动，是企业的核心应用。该类应用负载具有如下特性：规模大，交易频率高，服务质量要求高，需要峰值负载平衡，对可靠性、弹性及安全性通常有很高要求。
- ▶ 分析和高性能计算类：该类应用负载主要承担企业后台的大数据量分析和计算。该类负载具有如下特性：计算量密集，输入输出带宽占用量大，运行中占用大量内存，处理大量浮点数据，需要系统具有较强的伸缩能力。
- ▶ 数据库和交易处理类：该类应用通常用于后台数据存储、处理以及交易。该类应用负载规模较大，对服务质量要求高，大量使用内存，通常是响应式基础设施。
- ▶ Web、协同以及IT基础设施类：该类应用负载具有如下特点：高并发访问线程，具有较高的吞吐量要求，需要有较强的系统伸缩能力，对服务质量要求相对不高。
- ▶ 开发测试类：该类应用主要支持软件的开发测试活动，所开发的是前面所述应用。该类应用负载对系统资源的消耗通常随着开发测试的生命周期变化而变化，需要有较强的系统伸缩能力。

在梳理完企业的应用负载后，下面需要综合考虑应用负载组合的特性及对云计算的要求，据此判断该类应用负载是否适合转移到云计算平台上及采用何种转移战略。

关于某一类应用负载是否适合转移到云计算平台上，我们可以采用“投入-收益”矩阵来进行分析。“投入-收益”矩阵包括“投入”和“收益”两个维度，即“低投入高收益”和“低投入低收益”，以及“高投入高收益”和“高投入低收益”。根据应用负载组合的特性，将其按照实施云计算所需的投入和所获的收益按比例放入到该矩阵中，就可以得到如图3.3所示的应用负载组合分析图。位于“低投入高收益”象限中的应用负载是比较适合首先转移到云计算平台的。

关于一个应用应该放在如图3.4所示“投入-收益”矩阵的具体什么位置，取决于企业对于将该应用转移到云计算平台的投入和收益估计。需要指出的是，不同企业有自身的特

色，即使相同的应用，投入收益估计也不尽相同。图3.4中应用负载的“投入-收益”组合分析反映了一般企业的判断，仅供企业在做相应分析时参考。

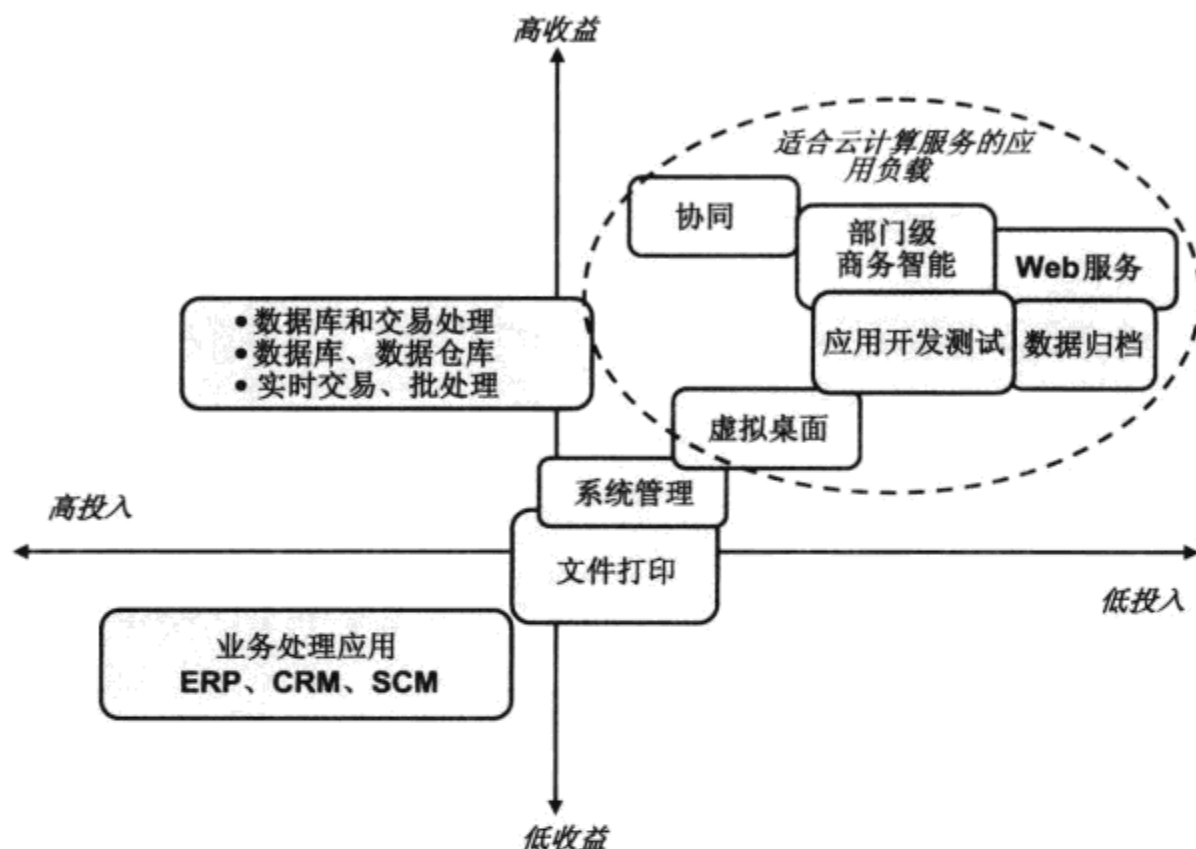


图3.4 应用负载的“投入-收益”组合分析

在确定了适合采纳云计算服务的应用负载后，我们需要回答第二个问题：这些应用将采用何种云转型策略，即如何从现有的应用转换为基于云计算的应用。为了回答这个问题，我们可以采用应用负载云实施策略分析框架来进行分析决策。该框架由两个维度构成，分别是应用负载的云消费模式和云适应性。云消费模式是指该应用负载采用何种云计算部署模式，主要包括公有云、私有云和混合云模式。应用负载的云适应性是指该类应用负载与云计算架构的兼容性或适用性，即指将该应用负载部署在某类层次的云计算平台是否需要改变原有的应用配置和应用流量。这种兼容性主要包括基础设施兼容、运行平台兼容，以及软件应用兼容等。

一个采用应用负载云实施策略分析框架的分析如图3.5所示。从中可以看出，与软件应用兼容性较好的应用负载适宜采用云迁移的实施战略，也就是说无须对应用负载进行显著的改造，直接将其迁移到兼容的云计算平台上即可。其他应用负载则需要采用云转型的实施战略，也就是说需要对应用负载进行显著改造，以保证其兼容云计算平台的基础实施和

运行平台。

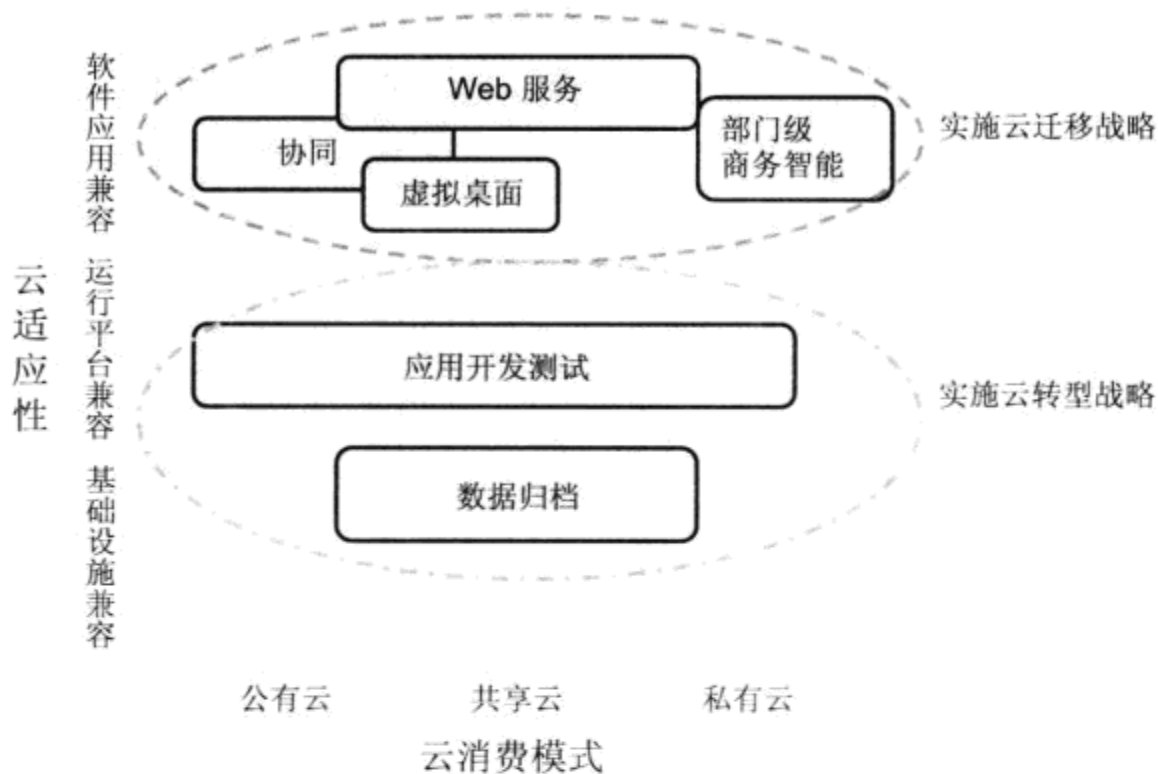


图3.5 应用负载云实施策略分析框架

通过以上分析，我们可以确定各类应用负载的云实施策略，即是否采用云计算服务，进而是采用云迁移还是云转型来实施云计算服务。由于云转型需要对应用负载进行显著的改造，在实施的过程中需要对改造的内容和方法进行评估与规划。

3.2.5.2 云转型的评估与规划

云转型即为对应用负载按照云计算平台的需要进行改造，使之能够运行在云计算平台上。为了实施云转型，首先应对转型过程中应用负载需要改造的内容和工作量进行评估。为此，我们要了解云转型的方法过程、评估因素及路线图的制定。

对某个应用负载进行云转型评估与规划的过程主要分为三个阶段，如图3.6所示。首先，收集云转型的需求，包括云转型的业务动机、应用负载的技术架构、云转型实施的计划、以及期望转型后具备的云计算服务能力等。其次，制定云转型的备选方案，根据该应用云转型的基本需求和计划以及该应用的技术架构，列举出各种可行的备选方案。备选方案包括云服务提供商、云服务类型、必要的转型活动、转型工作量等。最后，确定最终方案。根据对各备选方案进行综合评估，决定最终采用的云转型方案。

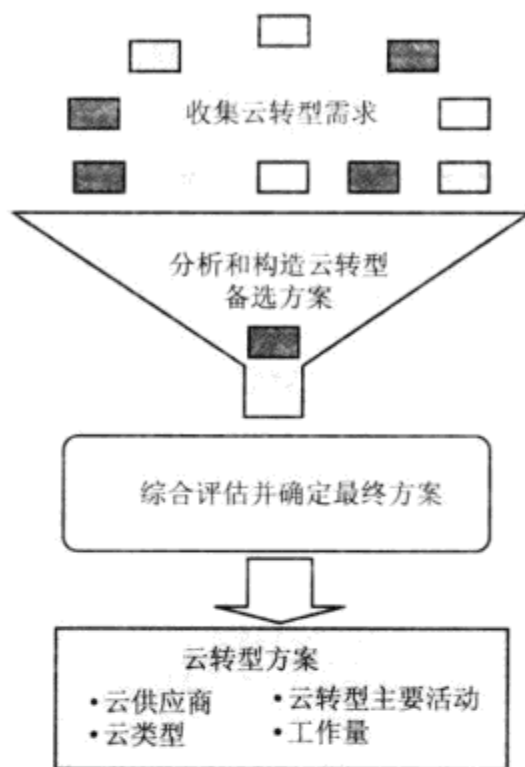


图3.6 应用负载云转型评估与规划过程

对应用负载进行云转型评估时需要考虑技术和非技术两方面因素。技术评估主要关注现有应用与未来云平台在基础设施、运行平台、运营支撑服务及业务支撑服务上的兼容性。在云转型评估的过程中，技术评估是重中之重。技术评估确保了云转型的可行性，同时又可以确定实施云转型的活动和工作量。非技术评估主要关注财务、法律法规、组织与人事、文化等能够影响转型的因素，它保证了云转型在非技术层面上可以顺利实施。通过对每一个备选的云转型方案进行技术和非技术两方面的评估，再综合考虑云转型的业务动机和转型约束，我们可以选择出合适的负载云转型方案。

3.3 企业云计算业务的实施

在确定了云计算的业务战略并且完成实施规划后，企业面临的问题就是怎样将业务迁移到云平台并确保业务的成功上线，这是实施云计算要面临的执行问题。如图3.7所示，与传统的业务模式不同，在基于云计算的业务模式中，业务和资源并不一定存在绑定的关系。在云计算中，每个业务系统未必独占涵盖从网络、存储、服务器等硬件资源到操作系统、数据库和应用服务器等软件资源。云计算业务的实施有其独有的特点，因此不能完全照搬以往IT系统实施的经验。事实上，云计算业务相比于传统的IT业务，实施变得更为简

- android与iphone及ipad开发书籍 -----持续不断更新中.....
- c、c++、c#语言pdf书籍及vip视频教程 c、c++、c#、vc等-----持续不断更新中.....
- delphi《书籍》及《视频》教程 -----持续不断更新中.....
- E网情深VIP系列视频教程 黑客破解菜鸟修练班，VB编程学习班，仿站学习培训，免杀培训，个人系统攻防系列教程，服务器搭建学习班，PHOTOSHOP平面设计班，基础制作论坛（论坛网站搭建），网赚系列教程，网站建设教程，网站漏洞基础，远程控制教程，软件破解班，脚本漏洞提权班
- IT9网络学院VIP系列视频教程 免杀培训班，VMware虚拟机，零基础学习C语言，网游外挂开发精品系列语音教程（外挂教程学习必备研修31课全），VB语言教程30课全，Delphi编程到精通，远程控制软件，加密解密班，网络安全与黑客攻防培训，从入门到精通完整系统化学习C++编程，从入门到精通零基础学习汇编，wordpress教程(个人博客系统49课全)，外行人做易语言盗号和钓鱼程序语音教程 [网址：WLSAM168.400GB.COM](http://WLSAM168.400GB.COM)
- Java书籍 -----持续不断更新中.....
- photoshop、CoreDRAW、AutocAD等图像处理书籍及vip视频教程 -----持续不断更新中.....
- powerbuilder书籍大全
- Visual Basic语言vip视频教程及pdf书籍 -----持续不断更新中.....
- windows、linux系统开发、系统封装等pdf书籍及VIP视频教程 -----持续不断更新中.....
- 《3DS Max》pdf书籍
- 《汇编语言》、《反汇编》及《调试》pdf书籍及vip视频教程 -----持续不断更新中.....
- 《电子书、电子书、还是电子书》pdf专题库 编程开发，家居美食，儿童益智，人物传记，增强记忆，快速阅读
- 信息系统项目管理师、网络工程师、系统分析师等软考类书籍
- 华中红客系列vip视频教程 脚本攻防培训班，源码免杀培训班，Css语言培训班，C语言，Dreamweaver网页设计，html网页设计培训班，PC安全班，php脚本语言培训班，VMWare虚拟机专题，webshell提权培训班，防站教程，零基础免杀培训班，刷钻速成班，脱壳破解班，外挂编写班，网络赚钱培训班，网站入侵培训班
- 外挂、驱动、逆向及封包视频教程 郁金香、独立团、夜猫论坛、天都吧、看流星论坛、一切从零开始等等
- 安全中国系列vip视频教程 易语言软件编程培训班，ASP.net网站开发项目实战培训班
- 我的收藏
- 按键精灵及TC脚本开发软件视频教程 -----持续不断更新中.....

当前位置： / 《电子书、电子书、还是电子书》pdf专题库 ←

文件名 ◆ PDF电子书专题库，内容详尽，每天不断更新！！

- 办公类软件使用指南
- 医学
- 历史人物传记
- 哲学宗教
- 外语资料（除英语外）（除英语外）
- 官场类小说
- 建筑工程类
- 情感生活类小说
- 政治军事
- 教育学习科普大全 [网址：WLSAM168.400GB.COM](http://WLSAM168.400GB.COM)
- 文学理论
- 智力开发、增强记忆、快速阅读技巧大全
- 社会生活
- 科学技术
- 程序编程类
- 经济管理
- 网络安全及管理
- 网赚系列
- 美食小吃烹饪煲汤大全
- 课外读物

本网盘内容太多，持续不断更新，发布各类视频教程、pdf书籍，包括破解、加解密、外挂辅助制作，易语言培训教程、编程语言、网页制作等等，教程及书籍仅用于学习，如用于商业或非法律用途的后果自负！

- OX Foxit PDF Editor ±â¼-°æË"ËùÓÐ (c) by Foxit Software Company, 2004 VIP培训教程，易语言黑月VIP视频教程，天½öÖAÖUÆA¹A¡£
- 都网易语言系列培训教程(100集全)，集中营易语言学习视频(80集)
 - 棉猴系列vip视频教程 gh0st远程控制源码讲解教程，套接字编程，DLL程序编写，键盘监听驱动程序编写，驱动基础教程，AsyncSelect模型QQ程序教程，C++语言入门基础，NB5.5源码分析教程
 - 游戏开发pdf书籍 -----持续不断更新中.....
 - 炒股投资pdf书籍及视频教程 短线高手系列，短线天王系列，操盘论道系列，翻倍黑马，看盘快速入门，庄家手法大曝光等等。 [网址：WLSAM168.400GB.COM](http://WLSAM168.400GB.COM)
 - 热门小说集中营 傲世九重天，网游之三国时代，武动乾坤
 - 甲壳虫VIP教程全集 asp教程，Delphi培训班，FLASH培训班，Java培训班，linux培训班，PHP培训班，源码免杀班，甲壳虫C++，脚本攻防班，免杀班初、中、高级班，破解班，源码免杀班，脱壳班，易语言培训班，无特征码免杀，网站架构培训班，外挂高级班，外挂初级班第1、2部
 - 破解、免杀、入侵、脱壳、攻防及漏洞分析系列VIP视频教程（80多部） 天草、黑客动画吧等等-----持续不断更新中....
 - 网站建设相关的pdf书籍及各种vip视频教程 -----持续不断更新中.....
 - 网赚、淘宝系列vip视频教程 网赚30天新人魔鬼训练，屠龙网赚团队vip课程，站长大学网赚视频（50课全），图腾团队日赚1000元竞价营销教程，屠龙团队淘宝宝贝卖疯系列，站群网赚系列，淘宝开店视频，红星挂机日赚10元，百万流量系列，漂流瓶圣手全自动挂机引，贴吧邮件定向营销疯狂成交量月入万元
 - 英语学习资料百科大全 不断更新。。。
 - 饭客论坛系列VIP视频教程 脚本入侵班，黑客之免杀教程，易语言教程，无线网络攻防教程，入侵教程，delphi系列教程，黑客基础入门
 - 黑客书籍 有关黑客、安全、加解密技术等等-----持续不断更新中.....
 - 黑手安全网VIP系列视频教程 DIV+CSS网页布局，Dreamweaver教程，flsah动画教程，photoshop教程，跟我一起学C++课程，抓鸡
 - 黑鹰、黑基、黑防、黑盾vip系列视频教程 破解提高班66课全，SQL注入，ASP注入教程，完完全全会抓肉鸡，脱壳破解教程50课全，提权班，C语言特训班26讲全，黑客脚本特训班，黑客工具特训班，dedecms仿站教程，VC编写远控30课全，网页美工特训班，木马免杀特训班，驱动开发技术VIP培训班，外挂破解等等。

- [电脑世界的通关密语：电脑编程基础].(杉浦贤).滕永红.扫描版.pdf
 - [程序语言的奥妙：算法解读（四色全彩）].(杉浦贤).李克秋.扫描版.pdf
 - [差错：软件错误的致命影响].(帕伯斯).邝宇恒等.扫描版.pdf
 - [算法之道（第2版）].邹恒明.扫描版.pdf
 - [O'Reilly：深入学习MongoDB].(霍多罗夫).巨成等.扫描版.pdf
 - [深入浅出WPF].刘铁猛.扫描版.pdf
 - [Go语言·云动力（云计算时代的新型编程语言）].樊虹剑.扫描版.pdf
 - [精通.NET互操作：P/ Invoke、C++ Interop和COM Interop].黄际洲等.扫描版.pdf
 - [编程的奥秘：.NET软件技术学习与实践].金旭亮.扫描版.pdf
 - [O'Reilly：学习OpenCV（中文版）].(布拉德斯基等).于仕琪等.扫描版.pdf
 - [Go语言编程].许式伟等.扫描版.pdf [网址：WLSAM168.400GB.COM](http://WLSAM168.400GB.COM)
 - [MySQL技术内幕：SQL编程].姜承尧.扫描版.pdf
 - [Tomcat权威指南（第2版）].(布里泰恩等).吴豪等.扫描版.pdf
 - [Ext江湖].大漠穷秋.扫描版.pdf
 - [IT名人堂·Oracle DBA突击：帮你赢得一份DBA职位].张晓明.扫描版.pdf
- Total: 77 1 2 3 4 5 6 >

[HTTP://WLSAM168.400GB.COM](http://WLSAM168.400GB.COM)

便，这也是云计算价值的体现。

下面将概要介绍企业在实施云计算业务方面所需的工作。云计算平台不同的层次，如基础设施层、平台层和应用层，带给用户使用的灵活程度各不相同，实施云计算的方式也有所差异。本节将分别说明这三个层次上实施云计算的方式。

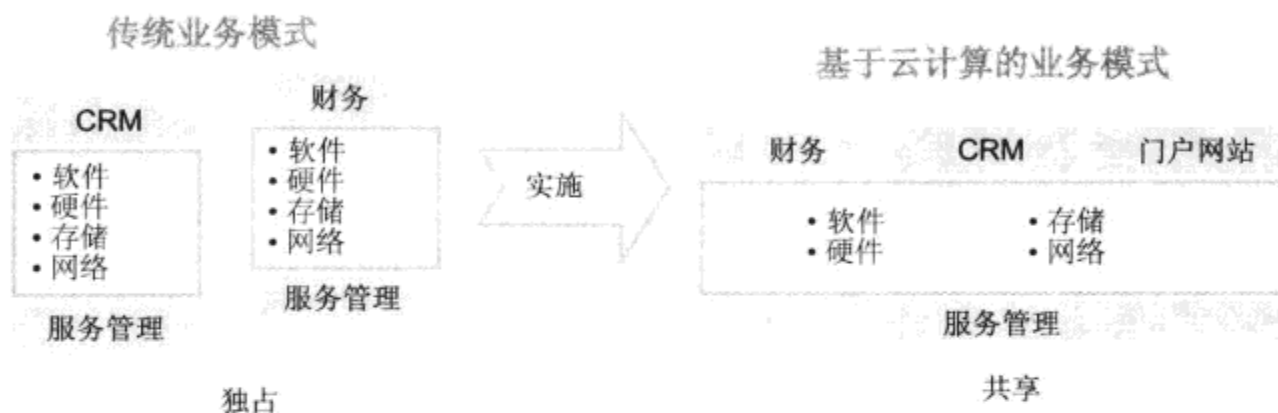


图3.7 传统业务模式与云计算业务模式的区别

3.3.1 实施基础设施层云计算业务

云基础设施层为用户提供灵活动态、低成本的基础设施服务，如网络、存储及计算资源等。利用基础设施服务，用户可以快速将传统的业务迁移到云计算平台，或者快速创建新的业务。目前主要的云基础设施服务提供商主要采用虚拟化技术将资源组织成资源池，以实现对资源的统一管理和调度。本节实施的云计算业务直接利用云基础实施层的服务。

如图3.8所示，实施基础设施层云计算业务一般包括下面几个步骤和活动：准备阶段、资源规划、创建业务模板、部署业务、安全配置、测试验证、发布上线和运行管理等。下面将逐次介绍。

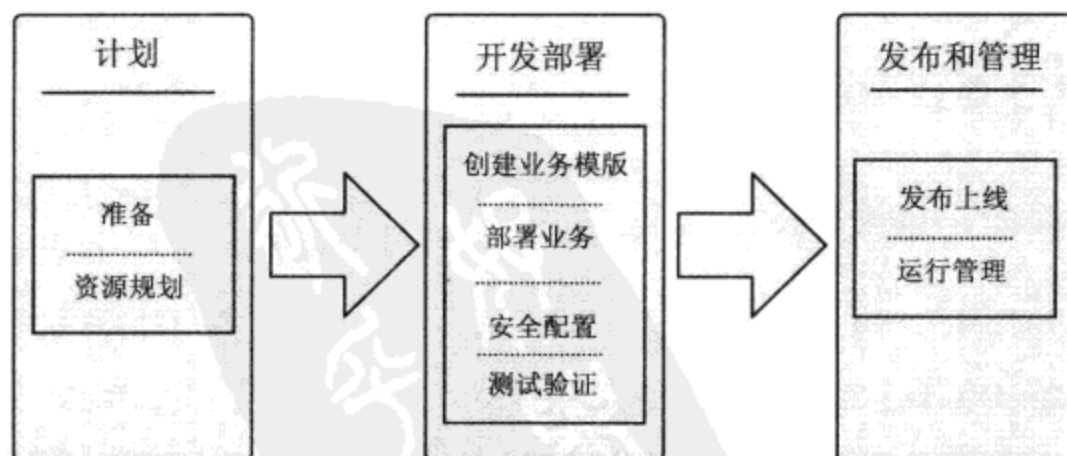


图3.8 实施基础设施层云计算业务的流程

充分了解云平台的功能和性能是实施成功的关键因素之一。云计算用户在准备阶段需要结合自己的业务需求，通过培训和学习等途径了解目标云基础设施层的功能和性能，如是否提供足够的可靠性，是否能够保证数据和应用的安全性，对资源的上线有没有限制，资源是否能够自动伸缩，为用户开发提供哪些应用程序接口（API）或者工具等。

明确了业务的需求及平台的功能与性能后，资源的规划便是接下来要考虑的问题，主要包括选择哪些基础设施服务，采用什么样的部署结构和拓扑来将业务实施在云中，每种资源的初始数量和将来资源随业务负载调整的模式等。

虚拟机模板是云计算业务在基础设施层的表现形式，模版实例化后即为运行的实例。虚拟机模板的选择和创建是最核心的步骤，不同的基础设施层提供服务的能力是不同的，如有的限制用户只能使用平台提供的虚拟机模板，不支持用户上传自定义的模板；有的支持用户自己提供自定义的模版。由于不同的云平台对虚拟机模板的要求有所差异，一般的做法就是从基础设施层服务提供商的网站下载标准的模板，然后进行定制，之后再转化为自己使用的模版并上传到云平台中。

用户在提交或者定制了自己的业务模版之后，即可利用云平台的部署功能创建并配置业务运行实例。云计算平台的部署功能应该支持自动化、大规模并快速部署。

部署完成后，用户可对业务运行实例进行测试和验证，然后对虚拟机的安全进行配置，包括防火墙设置、认证方式、数据传输安全等。所有这些工作完成后就可以将运行实例作为业务发布上线了。运行管理是贯穿从业务上线到终结整个生命周期的工作。运行管理可以利用云计算平台所提供的管理控制台来完成主要的日常运行和维护工作，从而有效地减轻业务系统的运维负担。

3.3.2 实施平台层云计算业务

3.3.2.1 平台层服务选择原则

云计算平台层为用户提供应用开发、运行和管理的环境。其中的运行环境要具备可伸缩性，能随着应用负载的起伏动态地调整资源供给，以确保应用性能的一致性。现在，越来越多的开发人员和企业正在考虑或者已经将应用迁移到平台层上来。同时，可供选择

的PaaS服务提供商也越来越多，如Google App Engine，Microsoft Azure，Salesforce force.com等。对于PaaS的用户来说，选择合适的PaaS服务是在PaaS上运行应用的关键。下面将介绍选择使用PaaS服务所应关注的几个重要问题。

(1) 要选择合适功能的PaaS平台，即关注PaaS提供的功能是否正好是所需要的。当前，有些PaaS提供通用的Web服务或者数据分析服务，有些PaaS平台针对特定的行业或者应用提供定制化的模块或者服务，有的侧重于应用开发，有的擅长应用的运行管理。用户需要根据自身的需要，选择与自身的需求和已有应用最为契合的PaaS服务。在这一点上特别需要注意到PaaS与我们所熟悉的应用开发工具或者管理平台的区别。例如，某些开发语言或者工具为了适应PaaS平台的结构和功能而受到剪裁；数据存储与运算处理通常是由分立的PaaS服务所提供的；事务（Transaction）管理机制为了适应可扩展性的要求而有所变更。在开发新的基于PaaS的应用或者把已有应用迁移到PaaS平台上的，上述技术方面的差异将影响到任务的难度和工作量，因此要预先做好打算。事实上，每一个PaaS服务商针对的是某一类特定的应用场景。以GAE为例，它关注高可扩展性的应用场景，为了适应高度可扩展性的要求，它对所支持的应用做出一系列的限制。有人估计，99%的应用选择GAE都是不明智的，对于剩下的1%，在清楚了解GAE的工作机制和限制后，能够高效地运行起来。

(2) 要选择成熟的PaaS平台，也就是选择那些经历了较长时间发展，所提供的核心功能都稳定下来，在一定时间内不会有关键性变动从而使得此前的开发和管理工作不会白白浪费掉的PaaS平台。判定一个PaaS平台是否成熟要察看多个方面的因素，核心是自己所需要的功能和性能是否会经历频繁的变化。例如，PaaS的API是否在不断更新，新版本的API是否修改了已有的功能，老版本的API是否依然可用；系统的限制是否在不断变化，这些变化是增多还是减少。

成熟的PaaS服务还体现在提供商的支持服务是否充分。例如，各类使用文档和用例是否清晰详实；各类支持渠道是否随时可得；用户所遇到的问题是否能够得到及时有效解答；是否有广泛的协定来规范服务供需双方在服务使用和保障方面的责任和义务，以及在出现事故的时候如何辨析责任并提供相应的保障和赔偿。事实上，PaaS平台在当前还处于初始阶段，即使大名鼎鼎的GAE也还是Beta版。因此，用户应该关注自己所需的功能是否足够成熟，此后的发展趋势对自己是否利好，而无须等到PaaS平台充分成熟之后才开始使用。当然，对于那些关键的应用，用户在考虑使用PaaS平台时

要采取慎重态度。

(3) 要选择运行稳定的PaaS平台，也就是有证据表明平台的运行如所宣称的那样符合服务等级的要求，应用性能、可靠性和可用性达到了协议的规定。如果一个PaaS平台在运行过程中不断动荡，中断服务的事件时常发生，所获得的服务性能起伏跌宕，这样的平台将严重影响用户应用的服务质量。用户应用在设计的过程中要考虑这些影响服务质量的因素，但PaaS平台要提供自身性能的评估并通过服务等级协定确定下来。用户需要时常监测以便确定PaaS平台是否确实达到了其承诺的服务等级。稳定运行与前述的是否成熟是不同的：是否成熟是从PaaS平台的功能变动角度来考察，而运行稳定是从PaaS平台的性能变动角度来考察。

(4) 要尽量避免被锁定。PaaS用户要根据PaaS平台提供的接口API编程实现自己所需的功能，这样应用才能够运行在相应PaaS平台上并利用PaaS平台的基础服务。如果PaaS平台提供的接口是仅为自己所支持的，那么采用这样接口的应用也就只能在该PaaS平台上运行。当用户因为服务质量或者价格的原因需要把应用迁移到其他平台时，采用特定API的应用在迁移的时候遇到的困难将更大。这就是所谓的锁定（Lock-in）。从PaaS提供商来说，锁定用户将有助于其保持持续的利润，而过于深度的锁定也会影响到用户考虑采纳其服务。PaaS平台采用开放的架构和接口将有助于减轻锁定的风险。当前的现状是各个PaaS平台的接口各式各样，还没有实现统一和标准化。标准化和互操作是PaaS平台在今后一段时间需要特别关注的工作，标准化有助于吸引并扩大PaaS的用户群和产业规模。从PaaS用户的角度来看，评估PaaS用户提供的API，考虑应用被锁定的风险和程度以及应用迁移的成本，是选择某个PaaS平台的重要因素。

总之，PaaS用户应该选择那些满足自己需要、服务可靠、容易迁移的PaaS平台，从而使自身应用得到恰当的服务和可靠保障。

3.3.2.2 平台层业务的实施

实施平台层云计算业务，就是指基于平台层环境开发并运行相应的应用，或者将传统的应用通过相应工具转换成能在平台层运行的应用。

实施平台层云计算业务的流程如图3.9所示，它包括的阶段和主要活动有：准备、资源规划、业务开发测试（或者迁移）、业务部署、测试验证、发布上线，以及运行管理等。

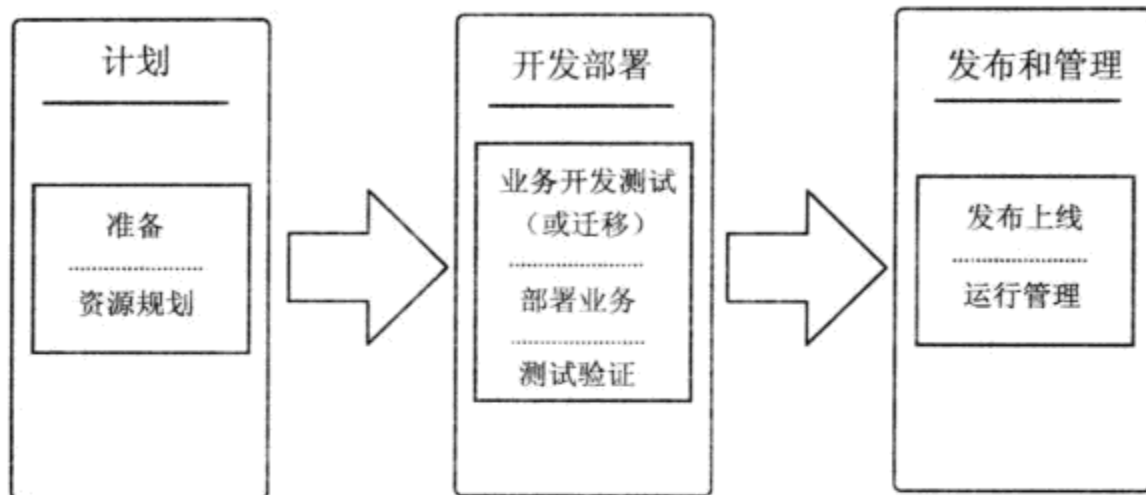


图3.9 实施平台层云计算业务的流程

平台层提供的核心功能是应用运行环境和相关基础服务，所以准备阶段的关键就是了解平台层的基本功能和特点，如支持什么样的运行环境，是否提供关系数据库服务，是否提供统一用户认证的服务，是否提供自定义报表服务，是否提供 workflow 引擎等。此外，也需要考虑平台层的相关资源要求，比如许可证的购买和管理等。

在了解了业务的需求以及平台的能力后，下一步便是应用的开发或迁移。在这里，我们可以利用平台的开发工具包和服务实现新的业务，或者利用平台提供的工具将传统的业务系统转变成云上业务。在平台层开发的新业务可能将面临与所在平台绑定紧密而无法迁移到其他平台的风险。用户需要平衡两方面的需要：是选择利用平台层提供的服务减少开发负担还是选择更强的跨平台兼容性。应用安全也是业务开发需要关注的方面，包括选择安全的访问协议，对数据进行加密等。

当业务开发完毕，在被真正部署到运行环境前要进行测试。这包括两个阶段，首先利用平台层提供的本地模拟测试环境进行测试，这个环境模拟了平台的绝大多数功能；其次是将业务部署到云平台上进行集成测试。在这里，部署的对象与基础设施层不同，平台层部署的业务系统只是应用程序，而不是虚拟机。平台上的测试主要测试业务应用同平台层服务的集成效果，如对认证服务、数据存储服务等进行集成测试。

测试完成后业务就可以发布上线了，发布时应选择域名以及发布的版本等。平台层的管理环境为用户提供了丰富的功能，比如统计应用的访问记录，存储和计算资源的使用报表，应用的版本管理，域名管理等。平台层云计算业务的运行管理应该充分利用平台层管理环境所提供的功能，从而减轻应用运行管理的负担。

3.3.3 实施应用层云计算业务

与前面两种层次的云计算平台相比，应用层提供给用户的是包括硬件到操作系统、中间件及上层业务的最完整方案。用户只需要选择合适的应用层服务提供商，通过云平台提供的工具进行定制就完成了云业务的实现。

实施应用层云计算业务的流程如图3.10所示，它主要包括准备、业务规划、业务选择与定制、安全配置，以及发布上线等。对于应用层服务的用户来说，准备和业务规划阶段是最关键的，因为平台提供的服务决定了用户所能具有的业务能力，用户很难通过开发获取新的功能。应用层服务也提供定制功能来适应不同用户的差别需求。如Salesforce.com提供的CRM服务，用户可以在标准服务的基础上基于自身业务的需求定制丰富多样的报表。安全配置是使用应用层服务的关键一环。通过配置限制访问业务的使用者或者网络范围等，用户可以保证业务的安全性。当定制和配置完成后，业务就可以发布上线，向使用者提供服务。

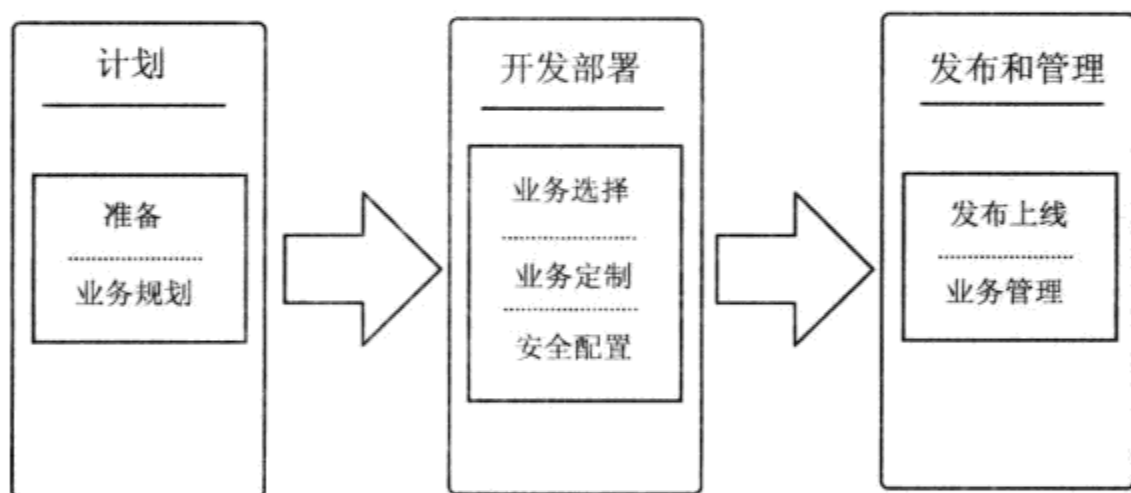


图3.10 实施应用层云计算业务的流程

本节所介绍的在三个层次进行云计算实施的流程是逐层简化的。企业在选择云计算业务实施平台的时候可以根据本节的介绍和自身业务的特点制定更为合理的实施步骤，以及在每一个步骤所需完成的事项和标准。

3.4 云计算提供商的业务模型

云计算提供商应该选择提供具有竞争优势的云计算业务。如何确定并实施云计算业

务，也就是确定合适的业务战略，是云计算提供商进行战略规划的核心内容。一般来说，业务战略是企业所处产业领域中的生存、竞争与发展之道。业务战略关心的重点是如何整合资源、创造价值，通过满足客户的需求来实现企业的盈利目标。对于云计算提供商来说，业务战略规划的根本是确定适合企业自身条件，具有竞争优势的业务模型。本节首先介绍云计算业务模型设计的基本内容，然后列举一系列可能的业务模型作为参考。需要注意的是，本节并不是关于企业战略规划的基本方法论（这样的方法论可以参考面向IT服务提供商的资料），而是以云计算业务模型为核心，突出云计算业务区别于一般IT服务的特点。

3.4.1 业务模型设计

云计算服务是一种新型业务，云计算提供商需要创新的业务战略和业务模型，其所制定的业务战略应该体现出对市场现状和发展趋势的深刻思考，真实把握云计算服务使用者的需求，同时也应该具有服务质量和成本上的竞争优势。在业务战略的指导下，位于市场中不同位置的云计算提供商会根据自身的特点和所看准的市场方向创造出各种各样的服务类型。云计算服务在运营方面具有的共性即称为业务模型。由于云计算是一种新型的IT服务，目前为止仅有少数成功的、可借鉴的云计算业务模型，而更多的业务模型还在酝酿和验证的过程中。另一方面，正因为云计算服务是新型的，还很难判断某个业务模型的创新是否成功。不过，在制订业务模型时，云计算提供商仍然应该遵循一些基本的原则。

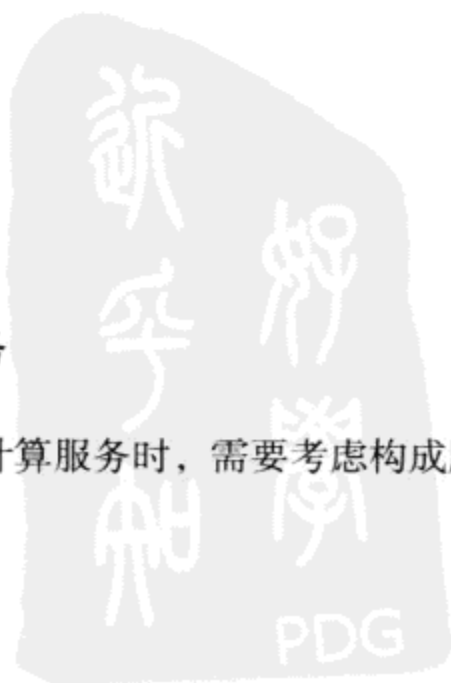
下面我们逐一分析云计算的业务模型所包括的四个基本方面，即：

- ▶ 如何选择所提供的云计算服务；
- ▶ 如何选择目标市场；
- ▶ 如何选择服务交付方式；
- ▶ 如何确保实现长期业务目标。

3.4.1.1 选择所提供的云计算服务

云计算提供商在选择所提供的云计算服务时，需要考虑构成服务的四个要素，即：

- ▶ 服务提供的内容；



- ▶ 服务流程和应用；
- ▶ 服务中间件与服务集成；
- ▶ 支撑服务的基础设施和设备。

云计算提供商需要分析自身在上述四个方面的特色，从而识别真正的竞争优势所在。一般来说，单个公司不可能同时拥有上述四个方面的能力。因此，它在设定业务模型时需要确定哪些部分由自身拥有，哪些部分应由其他的商业伙伴来提供。这样，云计算提供商之间将形成商业伙伴关系，通过伙伴之间的协作来实现对客户端到端的云计算服务。由于云计算服务的广泛性，可以预见基于伙伴关系的商业模式将成为云计算市场中的常态。云计算提供商之间的合作将体现在云计算的主要架构层次之间。如前文所述，这些主要的架构层次包括：软件即服务、平台即服务、基础设施即服务。

根据合作的商业模式，云计算提供者可以作为运营商，其业务直接面向客户；也可以作为支撑服务或者组件的提供者，为那些直接向客户提供云服务的运营商提供服务。后一类型的业务包括云计算服务的市场分析、云计算服务的开发、云计算培训等。针对云计算提供相应芯片和硬件的企业也属于后一类提供商。随着云计算服务的普及，对于云计算的咨询、执行和管理等相关服务的需求也会增加，从而成为对应提供商的业务。

3.4.1.2 选择目标市场

云计算业务的潜在市场前景广阔，云计算提供商需要不断开拓，让潜在的市场变为现实。在开拓市场时，云计算提供商应该根据自身的竞争优势和发展目标选定所服务的目标市场。当前，云计算提供商在选择目标市场时可以关注以下的市场方向。

- ▶ 大型企业将采用行业云。当前，许多大型企业考虑在企业内部建立云计算系统，其中一些企业也在考虑将自身的业务迁移到共享的行业云平台上。企业迁移到行业云的业务并不是其核心竞争力所在，如后台办公和合规操作等。由于同一个行业在运行和操作上具有相似之处，采用行业云可以提高企业的效率、节省成本、满足法律要求和行业规范。
- ▶ 对于中小企业而言，云计算服务可以降低他们的劳动负担并有效增加他们的收益。因为中小企业不存在已有的IT能力来阻碍采用新型的IT服务，他们最有可能首批全方位采纳云计算服务，而云计算提供商向中小企业提供云计算

服务也可以实现云计算的规模效益。

- ▶ 在新兴市场上的客户由于资源有限，更加看重云计算没有先期投资和可以灵活扩展的特点。面向新兴市场的云计算服务应该充分发挥云计算的这些优势。
- ▶ 云计算提供商或云计算运营商也将是云计算服务的目标对象。提供集成方案的云计算运营商需要采购合作伙伴的服务或者组件来提供综合的服务。例如，IaaS提供者往往和SaaS提供者结盟来共同提供切合客户需要的云服务。
- ▶ 移动设备、在线应用和其他消费服务受益于云交付模式所提供的众多新功能。云计算也可以为各种规模的企业在线支付、协作、自动化、应用开发和测试等提供机会。
- ▶ 在云计算业务发展的初期，大型企业喜欢建设内部的或者私有的云计算设施，它们仅采用少数类型的公有云服务。随着云计算业务的发展，大型企业在人力资源管理、采购、电子商务、数据仓库和数据存档方面对公有云服务的兴趣将日益增长，会考虑将公有云平台作为处理高峰负载的场所，或者将一些短期的项目转移到公有云平台上。对于中小企业而言，它们对公有云服务提供的网站托管、电子邮件、财务、费用管理和操作更有兴趣。云计算也适合具有移动办公需求的组织，如那些在新兴市场上拥有办公室的跨国公司。采用云计算服务，可以减少建立和管理分支机构的时间和负担。

3.4.1.3 选择服务交付方式

云计算提供商在运营云计算服务时需要关注云计算服务交付的关键要素。这些关键要素包括：能够更敏捷吸引客户的业务操作和财务报表；满足客户需求的服务定制；细粒度的定价方式和灵活的计费模型。为了实现云计算业务的运营，业务战略的运营模型需要定义外包模式、伙伴关系战略和相关的施行计划。许多提供商寻求战略伙伴关系来增强他们的能力，实现快速服务交付的能力。结盟和伙伴关系在云生态系统中往往是成功的关键。

明确的伙伴关系协议对于保护合作关系、减轻风险、合力承担投资需求和确保服务的连续性是必要的。随着云生态系统的演变，我们看到曾经分离的市场通过兼并，收购和结盟而逐步融合起来。在云计算生态系统的演进中，已有的参与者将变得规模更

大、功能更集成，而新的参与者也将不断出现。

3.4.1.4 确保实现长期业务目标

云计算提供商为了实现长期的业务目标，在业务模型上除了定义所提供的服务和目标市场外，还需要考虑定价模型、进入市场的方法、商业目的和价值主张。进入市场的第一批云计算提供商具有先发优势，而后来者的价值可以体现在客户关系、服务可靠性、技术创新、整合效率等上面。成功的云计算提供商应该发现并坚持具有竞争优势的市场定位。

云计算的一大特点是按使用付费的定价模型。在坚持基本的定价模型时，提供商也可以创造出变通的形式。例如，一些提供商发现它们的品牌和信誉允许它们能够基于所交付服务的价值来定价，而不是仅仅基于每小时使用率。这里的关键是如何衡量和决定价值，并且通过定价模型来体现这种价值。另一种可能是通过所消费的服务等级实现分级定价，对系统性能要求高的客户可以要求更高的服务等级，同时支付更高的价格。

此外，云计算提供商走向市场的方式各不相同。一些云提供商直接向最终消费者提供云计算服务；另一些提供商向前一类提供商提供产品和服务。在确定走向市场的战略时，向目标用户推销合适的服务很重要，需要根据它们的特点来定制相应的服务。云计算提供商可以对购买方的IT使用者和业务执行者进行了解，以提供有针对性的服务。新兴的云计算服务可以在数天内开发出来，因此开发新型服务的时候听取购买者的意见就显得很重要，它是云计算提供商实现差异化服务的前提。

3.4.2 业务模型示例

云计算提供商对自己的定位需要考虑多方面的因素，包括：

(1) 自身的特点和竞争优势。云计算提供商可能来自于广泛的领域，如通信、媒体和其他领域，包括设备制造商、网络提供商、内容提供商，甚至IT和应用外包服务提供商等。不同领域的提供商有各自深刻的行业背景，从而各有特点和优势。

(2) 所选定的业务模型。云计算提供商可以根据上一节介绍的确定业务模型时需要考虑的四个方面，选择最适合自己的业务模型。

(3) 服务提供的对象。云计算提供商可以选择向最终用户直接提供服务，也可以向

其他服务提供商提供技术和支持以实现其业务。

云计算提供商可以选择的业务模型多种多样，除了面向客户的IaaS/PaaS/SaaS提供商之外，下面列举出其他的云计算提供商类型以供参考。

- ▶ 组件供应商（向其他云计算提供商提供硬件、软件或专业的基于云的服务）：作为云计算提供商的供应商，具有帮助云计算提供商的能力。这些能力体现在或者使交付的产品具有差异性，或者提高了集成的程度，或者增加了云服务的安全性，或者提供增值服务减小了云服务廉价化的风险。在云计算的生态系统中，组件供应商可以在支持混合云、云的集成、提供特定行业解决方案方面有所作为。
- ▶ 基于云的IT外包服务提供商（面向客户提供基于云的IT基础设施、应用服务或者迁移辅助）：客户在使用云计算服务的过程中，或者迁移到云计算平台的过程中需要业务和技术方面的协助。这为基于云的外包服务提供商带来了商机。这类提供商因其专业的知识可以帮助客户获得云计算的便捷性，或者向企业提供定制的云计算服务。外包服务提供商可以与面向终端用户的云计算运营商合作，增加后者所提供服务的类型和附加值。例如客户关系管理、支付处理、招聘、会计、人事管理等企业或者组织都需要的业务，可以成为首批基于云的外包服务。
- ▶ SaaS增值服务提供商（提供特定行业或者辅助性的SaaS服务）：通常的SaaS服务提供商的业务针对一般性的使用者，而增值业务提供商的业务针对那些有特定需求的SaaS使用者。这类提供者在通用SaaS服务的基础上增加补充性功能，聚合现有功能形成新的服务形式，从而满足那些有特定需求的使用者，或者那些希望简化使用SaaS服务的客户。增值业务提供商可以从特定的行业开始，首先满足特定行业使用者的需求，利用其独特的行业经验开展业务。他们的价值在于能够形成针对某个行业特定需求的解决方案，从而简化行业使用者使用SaaS服务的难度。随着业务的发展，增值业务提供商可以逐步增加跨行业的解决方案，直至创造出新型的业务模型。
- ▶ IaaS增值业务提供商（提供满足公司在访问延时、数据安全或者其他方面特定需求的IaaS增值服务）：IaaS提供商以服务的形式将基础设施提供给客户，根据客户的使用率来收费。IaaS提供商还可以基于服务等级、定制要求和安全要求等因素制定

差异化的定价策略。随着这类业务成熟起来而变成行业标准，竞争将变得更加激烈，最终成为价格敏感型业务。新进的提供商需要通过提供增值业务来保持和改进其盈利能力。他们可以在数据安全措施、业务部署方式、资产管理、监控和其他部署业务方面提供差异化能力，并据此形成差异化的定价模式。这些增值的部分可以根据服务的价值，而不仅仅是服务的使用率来收费，高价值的服务能够获取高额的收益。提供增值型IaaS服务为这类提供商赢得了生存空间。

除此之外，还有其他类型的云服务提供商，如云技术支持和咨询服务提供商等。随着时间的推移，新型服务提供商也将不断出现。开放的云计算平台有助于竞争和创新，不断创造出新型的云计算服务。

3.5 云计算提供商的平台构建

为了提供云计算服务，云计算提供商需要构建云计算服务的交付平台和运维管理平台。上一节已经提到，云计算提供商之间可以建立合作和联盟的关系，从而提供类型丰富的云计算服务。在构建云计算平台时，提供商也可以充分利用这种合作和联盟关系。因此，有两种基本的构建云计算平台的方式：

- (1) 自己建设完整的云计算平台；
- (2) 利用其他云计算平台的功能定制新的云计算平台。

前者适合需要完整掌控云计算提供的运营商；而后者适合那些需要快速提供新型或增值型云计算服务的提供商。值得注意的是，后者能够提供的云计算服务受制于其基本的云计算平台的能力。例如，SaaS服务提供商可以将其业务运行在IaaS服务提供商的平台上，而IaaS服务的部署能力、网络能力等会影响到SaaS服务性能。当然，已有的IaaS服务可以使得构建SaaS服务更加方便快捷，从而提升了SaaS服务进入市场的速度。

本节关注云计算提供商自己建设云计算平台的场景。云计算平台实际上就是云计算服务运行和交付所需的所有硬件、软件、管理及产品的开发、运维、销售和批价计费等功能集合。云计算平台的建设就是建设一个面向云计算服务的数据中心。建设数据中心

的方法、流程和管理也适应建设云计算平台。但是，云计算服务又是一种新型的IT服务，在建设云计算服务也有一些特别需要关注的方面。

(1) 要考虑所提供云计算服务对数据中心基础设施和管理能力的需求，这要求合理的容量规划。云计算服务的一个特点是能够快速扩张，并需要动态伸缩。已有实例显示流量倍增的时长可以按天来计算，这说明云计算服务的流量在短时间内可能有显著增长。动态伸缩发生的周期以小时计，这就进一步强调了容量规划时具备容许资源需求剧烈变化的能力。

(2) 云计算数据中心的建设要充分考虑建设成本，并通过规模效应来降低云计算服务运行所需的成本分摊。这决定了数据中心的地址选择、建筑设计、电力供应、网络连接等都需要统筹考虑。一方面应该选择那些建筑成本低、能源价格低的位置；另一方面应该选择网络访问方便、能够容纳数据中心的规模、升级扩容方便快捷、电力供应稳定的方案。

(3) 云计算平台的硬件包括服务器、存储设备、网络设备和安全设备等。为了实现这些设备的灵活和动态管理，我们通常采用虚拟化技术来屏蔽物理设备的差异，向上层提供一致的管理接口。在这里，选择兼容的物理设备和虚拟化方案是关键。由于服务器、存储设备和网络设备的虚拟化方案各自独立发展，这些设备组合在一起时的兼容性值得特别关注。

相应地，云计算平台所涉及的软件包括虚拟机管理器、操作系统、中间件、业务软件和相关的辅助软件。对这些软件的管理涉及软件的安装、配置、升级及监控等。云计算平台可以充分利用虚拟化能力实现软件管理的高度自动化。例如，预先配置和优化的虚拟机模版可以加快计算能力和应用功能的速度；合理的软件升级系统可以实现相同系统的快速升级。

当云计算作为一种服务提供给用户时，服务的管理就成为运营的关键，这是云计算运维管理系统所应处理的。区别于一般IT运维管理系统，（云计算运维管理系统）支持对服务的自助管理。云计算的运维管理将在下一节中分析。

在使用云计算服务的时候，用户的数据自然地迁移到云计算平台中。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，保证数据的安全、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，即使在灾难发生时，也

能快速地恢复数据。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。云计算平台设计和建设时要充分考虑到用户对于数据管理的深切关注并给出对应的措施。

云计算平台是新型的数据中心，相应地，当前最新的节能和绿色技术也应该反映在云计算平台的设计和建设中。这样既可减少云计算运行的成本，又为环境保护承担相应的责任。例如，选择具有节能功能的硬件设备，提高数据中心的利用效率，合理设计和优化数据中心的制冷系统等。

总之，在云计算平台的构建中，要充分考虑云计算服务的特点，充分利用当前数据中心建设的最新成果，切实保证云计算平台的可靠性、灵活性、安全性和便捷性，有力保障云计算服务的成功。

3.6 云计算平台的运维管理

对于云计算提供商来说，当云计算平台根据规划和设计建设完成之后，下一步就是如何通过云计算管理平台进行运行维护的管理。云计算平台的运维管理是本节的主要内容。从云计算用户的角度来看，他们期望云计算服务具有一些重要特征，包括：

- (1) 服务的快速提供，最好是用户通过网页使用鼠标点击，就可获得所需要的服务；
- (2) 服务的动态伸缩，根据服务的访问量，管理平台能够自动调整支持服务的资源，从而保障服务的质量；
- (3) 灵活的定价策略，根据使用率收费。从服务提供商角度来看，他们希望自己的服务能够支持大量用户，能够适应用户的各种特定需求，能够快速开发出新的服务。

云计算服务的这些特点对运维管理系统提出了明确的要求。本节前三小节将介绍以上内容。

除此之外，云计算的服务模式是将数据和运算能力托付给远端的云计算平台，云计算用户仅通过网络访问云计算服务，用户失去了对数据和计算能力的实际掌控，因而信息

安全成为影响用户选择云计算最为重要的因素之一。信息安全管理是云计算运维管理的重点，本节最后一小节将介绍云计算的信息安全管理。

3.6.1 运维管理的目标

云计算运维管理的目标是：可见，可控，自动化。下面分别介绍这三个目标在云计算运维管理上的体现。

所谓“可见”，是指给用户和管理人员提供友好的界面和接口以便他们能够操作和实施相应的功能。当前的云计算系统普遍使用图形界面或REST类接口。通过这些界面或接口，用户可以提交服务请求，用户和管理人员可以跟踪查看服务请求的执行状态，管理人员可以调控服务请求的执行过程和性能表现，服务质量与资源使用状况的统计也可以通过直观的图表形式展现出来。

所谓“可控”，是指在运行管理的过程中整合人员、流程、数据和技术等因素，以确保云计算服务满足合同约定的服务等级，保证云计算提供商提供服务的效率从而维持一定的盈利能力。可控性关注的方面包括：根据最佳实践经验响应用户的服务请求并确保服务过程符合组织流程，确保服务提供的方式符合公司的运营政策，实现基于使用的计费管理，实现符合用户需要的信息安全管理，实现资源使用的优化，实现绿色的能源管理。

所谓“自动化”，是指云计算服务的运维管理系统能够自动地根据用户请求执行服务的开通，能够自动监控并应对服务运行中出现的事件。更进一步，自助服务是自动化在用户订阅和服务配置方面的体现。在实现“自动化”的过程中，需要关注的主要方面包括：自助服务的方式和自动化的服务开通；自动的IT资源管理以实现优化的资源利用；根据用户流量变化实现服务容量的自动伸缩；自动化的流程以实现云计算环境中的变更管理、配置管理、事件管理、问题管理、服务终结和资源释放管理等。

为了达到云计算服务运行管理的上述目标，云计算提供商需要建立相应的运维管理系统。运维管理系统的功能应该从云计算管理的目标出发，充分考虑云计算服务和计算资源的特点。例如，虚拟化资源可以实现灵活而细粒度的调度，基于SOA的服务架构和标准的服务接口支持业务的灵活编排和调用，自动化技术保证管理流程的快速高效等。运维管理系统的核心管理对象是云计算服务本身，它围绕云计算服务从开通到终结的

整个生命周期展开工作。

从IT管理技术的发展来看，云计算的管理也突破了传统的IT管理理念。传统的IT管理关注资源的管理，从底层资源的角度出发来保障业务和性能。云计算首先关注的是服务本身的性能，需要从服务性能的角度来调整和优化支持服务的资源供给方案。因此，云计算的管理是由底向上和由上到下的管理理念的结合。云计算的管理应该考虑到基础设施资源和技术的发展，业务特征和运维服务等因素，建构标准的、开放的、可扩展的云计算管理平台。

3.6.2 运维管理的核心

以云计算服务为核心的运维管理，体现了云计算环境中新型的管理理念。云计算服务的管理集中体现在对云计算服务生命周期的管理。

服务的生命周期在IT服务的标准ITIL v3中有明确定义。ITIL v3的核心架构是基于服务的生命周期。服务的生命周期以服务战略为核心，以服务设计、服务转换和服务运营为实施阶段，以服务改进来提高和优化对服务的定位及相关的进程与项目。具体到云计算服务，这些阶段依然是必要的。服务战略在前文中已经涉及，服务改进是服务运维过程中所关注的方面。在服务的实施阶段，云计算服务可以分为如下子阶段：服务模版定义，服务产品注册，服务订阅和实例化，服务运行和服务实例终结，如图3.11所示。

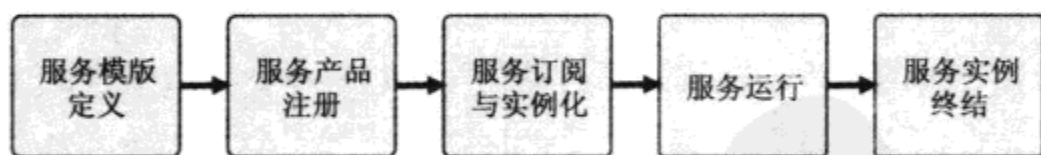


图3.11 云计算服务的生命周期

云计算服务生命周期的第一个阶段是服务模版定义。服务模版用于捕捉一类云计算服务在执行和管理时所需的数据和方法。一个服务模版包括两个部分，一是服务拓扑，二是管理操作。服务拓扑定义了一类服务所涉及的功能实体及这些实体之间的关系，而管理操作定义了执行服务过程中伴随的管理动作。

开发者将云计算服务模版定义好之后，就可以将它注册到云计算管理平台。管理平台

通过服务目录的形式将已注册的服务产品显示给使用者。在注册的过程中，管理平台会检查服务模版中所定义的功能实体和管理操作是否可行，以保证后续的服务实例化过程不会因条件的不具备而失败。

服务使用者通过浏览服务目录，选择并订阅相应的服务产品。这个过程可以通过自助服务界面进行，从而使得服务的订阅更方便快捷。在服务订阅的过程中，管理平台可以收集用户对服务的配置参数，以使生成的服务更贴近用户的需求。订阅服务的请求发送到管理平台后，管理平台调度对应的资源，快速部署并激活相应的服务，使用者即刻就可以访问相应的服务。这个过程可以通过流程引擎自动执行服务实例化流程来实现。服务实例化过程（即服务上线过程）应该是衡量云计算服务等级的一个重要方面，它由一组度量参数，如上线时间等来刻画。

实例化后的服务处于产品运行阶段。云管理平台在此阶段需要监控服务的运行状况，根据用户的配置变更服务的执行。通过测量服务的运行性能和资源使用状况，云管理平台应及时发现并解决运行过程中出现的故障和问题，优化服务的运行，在保障服务性能的前提下提高资源的使用效率，从而节省用户的费用。对大量运行阶段的服务实例进行监控和性能优化是云计算管理平台的重要功能，不同云服务平台在这项功能上的差异将直接影响到云计算提供者的赢利能力。

当云计算使用者不再需要某项服务时，可以请求终止该服务。支撑该服务的计算资源将被云计算管理平台回收以供后续的其他服务使用。服务的终止可以在约定的服务终止时间到达时自动执行。服务终止过程中的一个重要问题是要保证用户的数据不会受到损失，或者不会存在安全隐患。管理平台需要提供相应的流程和措施来保证这一点，例如删除数据前需要用户证实数据是否有用，提供可靠的数据备份和归档服务，删除数据时确保分散在不同位置的相同数据都被删除等。

在云计算服务的生命周期中，服务的自动管理可以大展身手。当用户提交了服务请求，自动管理可以根据服务模版实例化相应的服务。服务运行过程中可以通过自动监控和资源调整来保证服务的性能。在服务终结过程中，服务数据的保护和资源的回收可以根据设定的规则自动进行。服务自动管理依赖于虚拟化技术和标准化技术。虚拟化资源可以被管理系统灵活调度，标准化的接口可以屏蔽各种资源与服务的异构性，使得调用简单可行。

3.6.3 运维管理的平台

云计算管理平台是云计算提供商开发的，运行云计算服务的控制台，是云计算服务管理人员监控、管理、分析和优化云计算服务的重要工具，是支撑和保障云计算服务的信息化架构。图3.12给出了云计算管理平台的主要层次和功能。



图3.12 云计算管理平台的主要功能

如图3.12所示，云计算管理平台关涉三类参与角色。一类是云计算服务的开发者，通过云计算管理平台的开发者门户来开发注册云计算服务；一类是云计算服务的提供者，通过云计算管理平台运营云计算服务，在满足客户需求的同时获得对应的收益；第三类就是云计算服务的使用者。使用者通过云计算服务满足其需求。云计算使用者可以通过网络来访问服务，也可以创建自己的IT系统通过应用接口来访问服务，还可以通过其他合作伙伴的云计算服务来消费另一个云计算提供者的服务。云计算服务是这三类角色业务的核心，而云计算管理平台是这三类角色参与云计算服务的媒介。

云计算管理平台在实施管理时通过不同的管理功能来运行并保障云计算服务。这些管理功能可以分为三个层次：

- ▶ 业务支撑服务即面向客户服务和市场营销的支撑功能，管理用户数据和服务产品；
- ▶ 运维支撑服务即面向资源分配和业务运行的支撑功能，保证业务的快速开通和正常运行；
- ▶ 管理支撑服务即面向人力、财务、工程等企业管理的支撑系统，保障提供者企业的正常运转。

由于管理支撑服务是一般IT服务系统所共有的，我们在本节集中介绍业务支撑服务和运营支撑服务与云计算密切相关的方面，以便对提供云计算服务的特点有所认识。需要指出的是，三类支撑服务概念最早出现在电信服务的支撑系统中，后被广泛借用到支撑IT服务的系统中。

业务支撑服务提供产品目录和订阅管理，这是用户直接接触的部分。业务支撑服务还能够收集用户的概要数据，通过分析并定制出贴近用户特点的服务界面和产品推荐，简化用户的使用过程。自助服务界面，是云计算的特色之处。用户通过自助服务界面就能够实现对于服务整个生命周期的管理，包括产品选择、服务订阅、服务部署、运行监控，直至服务终结，以及此过程中所发生的费用计算和缴付操作。

云计算服务的另一个显著特征是服务等级管理，即SLA（Service Level Agreement）管理。服务等级协定是服务提供者和使用者的关于服务提供质量的协定，直接与服务的定价相关。服务等级协定所关注的性能指标与所提供的服务密切相关，通常使用的指标包括响应时间（Response Time）、吞吐量（Throughput）、可用性（Availability）等。业务支撑系统向用户提供服务等级报告，以使用户能够随时了解服务运行状况。而下面提到的运营支撑系统实施服务等级管理，从而实现服务等级协定所要求的服务性能。

运维支撑服务关注服务的开通和服务的保障，它通过对资源的调度和管理来实现对服务运行的支持。对于云计算来说，服务开通涉及服务模板管理、虚拟镜像管理、服务请求管理和部署管理等。服务保障的管理包括：配置管理、变更管理、事件管理、问题管理等。知识资产和软件许可证管理是云计算服务管理的一项重要内容，支持灵活快捷地获得业务运行所需的软件和资产。运维支撑服务的另一类重要功能是对服务性能的管理，即服务等级的监控和保障，通过监控资源的利用和服务性能表现，采用自适应调节的方式来满足服务等级的要求。

对于云计算用户来说，云计算服务运行在远端，他们考虑采纳云计算时的主要顾虑是信息安全和**服务连续性**。云计算的提供者应该采用一系列措施，使云计算服务的安全性和可获得性都要相当于甚至优于用户自己部署和管理相应的业务。从服务连续性来说，影响因素包括传输服务的网络通道和运行服务的云计算环境。云计算提供者应该将云计算环境放置在网络带宽充足有保障的地方，例如具有两个或者以上的不同网络接入商，而这些接入商的**网络基础实施是尽量不重合的**。在云计算环境内部，应该有足够的冗余措施来保证数据和服务的可获得性。例如，数据应该存放在多个位置，这些位置同时受到损害的概率足够小；计算任务能够快速地从一个受到损害的节点迁移到另一个正常的节点，例如，支持虚拟机的实时迁移。服务连续性措施应该在服务等级中定义且与计费策略直接相关。运营支撑服务是保证服务连续性的重要位置，特别是云计算环境中的服务连续性。

传统的IT系统管理关注的是IT系统的组成单元，如果这些组成单元是正常工作的，那就认定基于这些组成单元的应用服务是正常工作的。人们通常将服务的可获得性与相关的系统可获得性联系在一起。对于云计算来说，使用者购买的是服务能力，如计算容量、吞吐量等，而不是特定的系统单元，如特定的物理机器、网络设备等。同时，提供服务的资源通常是虚拟的，与特定的系统单元并不是绑定的。因此，在服务等级协定中规定的性能指标与云计算交付平台的组成单元并不是简单对应的。组成单元的可获得性并不简单等同于服务的可获得性。对于服务等级的测量也不是仅仅来自于对于系统组成单元的监控数据。有鉴于此，模拟测量和统计分析将成为云计算性能评定的重要方法。例如，可以通过发送并测量合成的服务请求的性能，来统计推断真实服务的性能。云计算服务的SLA管理将从关注被管的IT设备的性能转而关注用户体验到的性能。

云计算管理平台的根本目的是保障提供云计算服务的**服务质量和成本效益**，从而实现云计算的经济优势。管理平台应该通过一系列的管理工作流程来提高其自动化程度，从而尽量实现零接触（Zero Touch）管理，减少提供商的人力成本，提高管理人员的劳动效率来支持或者增强提供商的盈利能力。

为了提高业务支撑服务和运营支撑服务自动化水平，云计算管理平台需要采用一个强大的流程引擎，它可以编排并运行定制的服务流程，支持端到端的资源管理以及服务管理。下面简单的列出一些流程引擎应该具备的一些关键能力：

- ▶ 支持可视化的流程制定；

- ▶ 支持与使用者和被管服务对象的交互；
- ▶ 支持服务的生命周期管理；
- ▶ 能够监控和分析流程执行状况；
- ▶ 能够模拟并测试流程的行为；
- ▶ 支持多用户并具高度可靠性。

事实上，随着云计算服务市场的发展，云计算运维管理将成为云计算提供商之间的竞争点。支持管理大规模、多样化的云计算服务并具有高度自动化能力的云计算管理平台将为云计算提供商带来强大的竞争优势。

3.6.4 平台信息安全管理

云计算提供商应该采取相应的措施来保证云计算服务安全可靠。本节从云计算提供商的角度来看在保障信息安全方面所应采取的控制措施。由于安全管理和控制措施的选择应该基于风险评估的结果，下面将从介绍云计算服务风险评估的方法开始，然后列举一系列安全控制措施供读者参考。

3.6.4.1 云计算服务的安全风险评估

1. 安全责任划分

云计算服务在运行过程中，影响安全的因素涉及三个方面，即云计算的用户、网络通道和云计算提供商。一般来说，网络通道不应该假设是安全的，网络通道中数据可能被截取、查看和篡改。网络通道的安全措施已经有大量的研究，云计算服务应该充分利用网络安全的最新成果。另一方面，云计算用户和服务提供商在保证信息安全方面都具有责任，而责任的划分视业务模式的不同而不同。

举例来说，对于IaaS业务来说，云计算提供商应该保证云计算基础设施直到资源虚拟化层的安全性，而用户应该负责在所获得的虚拟化资源基础上进行计算和数据处理的安全性。对于PaaS业务来说，云计算提供商除了对应IaaS层次的安全措施之外，还应该保证提供平台服务的中间件和相关基础服务的安全性；而云计算的用户则应该负责自己的应用在

运行过程中应用和数据的安全。对于SaaS业务来说，云计算提供商所负责的安全措施应该涵盖提供应用所需的从下到上所有的层次；而云计算用户所承担的安全责任最少，但也应该掌握应用使用过程中的安全措施，如用户账号、数据传输等相关的安全性。因此，恰当划分用户和提供商的安全责任，是云计算服务在实施过程中安全保障的首要任务。

在明确了应该承担的安全责任之后，用户和提供商应该采取对应措施来履行这些责任。对于提供商来说，他应该以明确的方式告知用户自己所承担的责任和采取的措施，并且以有效的方式证明自己达到在安全方面的承诺，以使用户接受其提供的云计算服务。同时，提供商还可以通过提供工具和规范的文档告知用户应该采取的安全措施，从而吸引更多的云计算用户。对于用户来说，在选择云计算提供商时应该评估提供商的安全措施，验证安全措施的有效性，将安全措施作为自己选择提供商的一个重要依据。

2. 风险评估

在提出恰当的安全措施前，应该针对云计算系统和服务进行风险评估。所谓的风险评估就是根据信息安全的技术和管理标准，对信息系统及其在传输、处理和存储信息过程中的信息机密性、完整性和可行性等与安全相关的属性进行评估的过程。云计算平台作为一类信息系统，它的安全评估应该遵循一般信息系统的评估方式，并充分考虑云计算服务相关系统的特殊之处。现在有许多风险评估的方法，这些方法有大致相似的评估步骤。其中NIST在《NIST SP800-30: 信息技术系统风险管理指南》中提出的风险评估步骤（如图3.13所示），被广为采用。

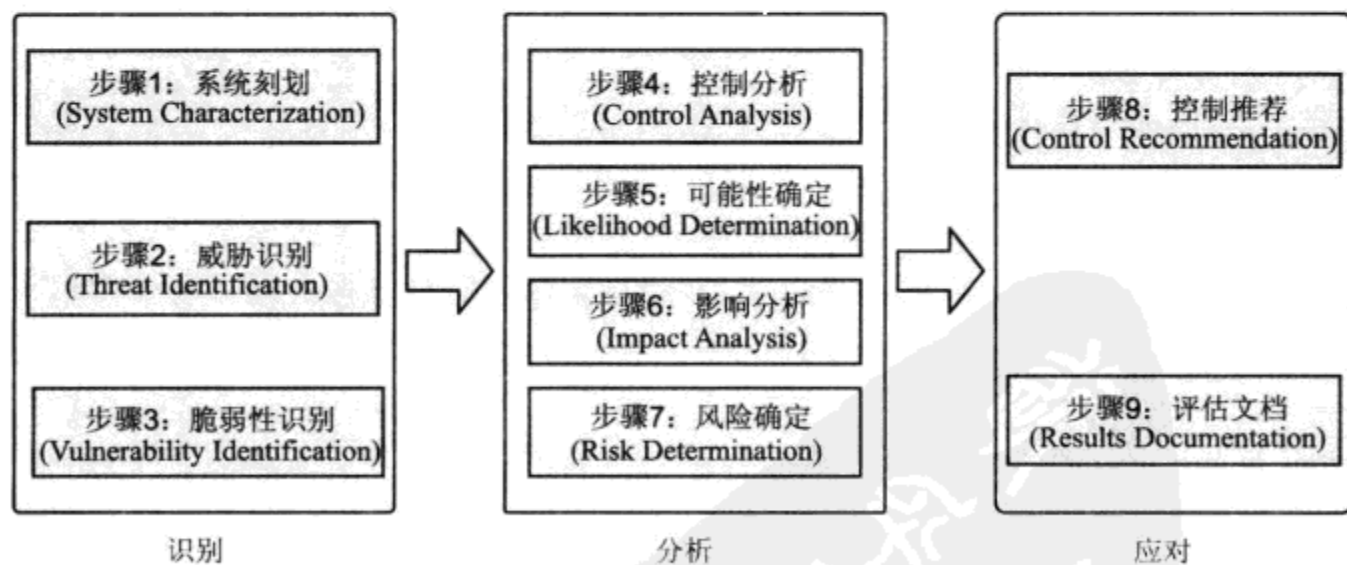


图3.13 NIST SP800-30中提出的信息系统风险评估步骤

对于云计算服务相关的风险评估来说，图3.13所示步骤1“系统刻画”应该充分考虑

到云计算系统的特殊性。这些特殊性表现在所采用的技术、系统架构、涉及的参与人员、业务的部署方式等方面。从技术角度讲，云计算广泛采用了虚拟化技术、面向服务的架构、Web应用和服务、加密技术等。这些技术的安全因素也自然而然地进入到云计算安全考虑的范畴。从云计算服务的架构来说，这是一种通过网络提供的服务，服务内容可能是IaaS、PaaS或者SaaS服务模式，或者是这些服务模式的组合。服务模式的不同从而确定的安全责任不同也会影响到风险评估。网络传输的安全风险应该纳入风险评估的考虑。

从涉及的参与人员来看，云计算作为一种新的服务提供模式，服务的开发者、提供者和使用已经紧密地联系在一起，他们之间的互动通过自动化的流程和自助服务界面而快速进行着。安全风险将会快速地在不同参与者之间流动，从而产生更为广泛的影响。从业务部署方式来说，一般有四种基本的部署方式，即公共云、私有云、介于公共与私有之间的行业云以及前述类型混合存在的混合云。行业云是指那些跨越企业边界、由一个行业或者社区共同使用的云服务。很显然，部署方式的不同影响到安全边界的确定及威胁方式和危害程度。前面仅仅列举云计算相关的几个方面，根据云计算服务系统的不同，这些方面可以进一步具体化，甚至还有新的因素值得考虑。

图3.13中所示的后续步骤也应该结合云计算系统的特别之处进行，从而找出影响云计算服务安全性的关键因素，并确定有效的应对措施。具体内容不在此赘述。

下面以“安全实施”为例，将NIST SP800-30的风险评估应用起来所需考虑的主要内容概括如下。

(1) 评估和文档化跟安全相关的组织文化。具体来说，一般包含：a) 评估目前同行业及本组织具体需求以建立安全基础架构的最佳实践；b) 如果交易机密是企业竞争优势的核心基础，则必须考虑安全相关的依从性；c) 如果出口控制对企业经营数据有相关要求，则必须据此考虑数据相关的安全依从性；d) 评估每个跟云相关的应用部署情况，在这个过程中，与云计算供应商和安全基础架构供应商的协商特别重要。

(2) 列出云计算实施项目设计的安全特征优先级。

(3) 创建并维护跟不同云应用需要的不同云安全特征和等级的策略和流程。需要注意的是，这些策略必须有效识别云环境的安全风险，同时，这些策略必须进行维护以反映当前的风险；这些策略必须提供有效的评估标准以监控风险方式的频率，这些评估标准一般需跟企业所处的行业法规相关；这些策略必须责任分明；这些策略必须提供风险发生时

推荐的应对措施。而应对措施必须包括关键行动、责任人、包含具体时限的申述流程。

(4) 与云计算项目实施团队高层主管协同工作，以了解云计算项目的需求并得到相应支持。

(5) 实施一个涉及全企业的安全培训计划以获得企业不同员工理解及支持，应该确保所有管理型员工完全所需的培训，确保云安全相关策略能够发布在这些员工工作的主要场所。

(6) 实施一个安全系统以确保安全状态和反差状态的透明度。

(7) 建立审计流程。

(8) 实施一个强制性的模式以确保事件及事件责任人的有效沟通。同时，建立有效的提醒机制以推动这些有效沟通尽可能实时地发生。

3.6.4.2 云计算服务的信息安全控制措施

云计算提供商在考虑云计算安全措施时应该充分了解相关的最佳实践，并结合自身特点选择恰当的措施。相关的业界标准或者最佳实践是提供商可以参考的重要资料。

事实上，当前已经有一些组织和机构针对云计算服务提出相应的安全措施，其中之一即为云安全联盟（CSA: Cloud Security Alliance）所发布的“云计算关键领域的安全指南”。CSA不是一个标准化组织，而是由云计算业界所组成的一个松散联盟，其目的是定义云计算的使用案例和指南，推广云计算最佳安全实践，提供使用云计算相关的教育，以帮助提升所有形式云计算的安全保障。CSA在2009年4月发布了云计算安全指南1.0版，而在2009年12月发表了安全指南2.1版。

下面以CSA安全指南2.1版介绍作为云计算提供商可以参考的安全措施。需要注意的是，CSA的安全指南不仅仅针对云计算提供商，用户在使用云计算服务时也可以参考该指南或者其他类似指南。

CSA的安全推荐可以分为与管制相关和与操作相关的两大类安全措施和实践。这两类又进一步分为十二个不同的领域。简单来说，跟管制相关的领域主要讨论和处理云环境中战略和政策上的安全问题，而操作相关的领域主要讨论和处理实际运营当中的问题和技术实践。下面逐一对这两类十二个领域进行简单介绍。

1. 管制类安全措施和实践

(1) 管制与企业风险管理 (Governance and Enterprise Risk Management), 主要包含与企业或组织在管理和度量风险的能力相关的内容。比如如何评估云服务提供商所带来的安全风险, 如何确定企业或组织以及云服务提供商在敏感数据保护上各自的责任等。简单来说, 为了维持有效的信息安全管制、风险管理和合规性管理, 企业或组织应该识别并实施合理的组织架构、流程和控制措施。

(2) 法律和电子证物 (Legal and Electronic Discovery), 云计算作为新生事物, 将带来新的潜在法律问题。此领域包含相应的措施以规避风险。随着云计算业务的发展, 云计算业务涉及多个参与实体的互动, 并将带来新的法律风险, 在规范云计算商业活动的法律、法规逐步增加并完善起来之前, 云计算提供商应该采取措施提供用户或者法律所需的明确的服务提供证据, 并且证据是可信的。

(3) 合规与审计 (Compliance and Audit), 主要内容为如何在云计算环境中满足各类法律、法规和相关审计的要求。一般来讲, 云服务合同中需要明确提供商和使用者在满足合规和审计过程中的责任, 并通过合同来清晰表达责任的划分。合同中还应确定提供证据的内容、时间、方式和证据取得的方法等。

(4) 信息生命周期管理 (Information Lifecycle Management), 主要关注云中的数据管理。主要包括设定数据访问控制, 设定数据加密方式, 验证数据在生命周期的各个阶段是否满足安全的需求等等。另外, 此领域也包含有一些其他机制的阐述, 主要用来保证云服务的用户在失去对物理设备的控制之后如何对数据进行管理。

(5) 迁移与互操作 (Portability and Interoperability), 云计算平台中的用户根据自身的需要在云计算平台之间迁移的情况并不鲜见。用户甚至有可能从云服务平台迁移至传统的服务平台。此领域提供了一些指南, 主要是关于用户如何评估迁移的风险和难度, 以及云计算提供商如何确定相应的迁移方法和流程。另外, 云计算提供商之间的互操作问题, 也是此领域所关注的内容。

2. 操作类安全措施和实践

(1) 传统的物理安全、业务连续性和灾难恢复 (Traditional Security, Business Continuity and Disaster Recovery), 主要关注云环境如何影响传统的物理安全、业务连续性和灾难恢复措施。除了可以借鉴传统数据中心的相关最佳实践之外, 此领域也讨论云计算

所带来的额外安全风险和措施，比如现场参观、信息披露、合同约定，以及有效的工作责任划分，限制工作人员访问用户数据等。

(2) 数据中心运营 (Data Center Operation)，主要包含 (云服务提供商的) 数据中心的架构和运行方式的相关安全措施和指南。比如，数据中心如何在不同的层次上进行功能划分，如何进行软件的补丁策略管理，以及如何满足安全的需求以及合理的集成等。总之，此领域主要用来使平台提供商的服务与用户相关需求能够更加平滑和一致地集成，用户也能根据自己的需求来对提供商进行选择。

(3) 事故响应、通知和补救 (Incident Response, Notification and Remediation)，主要关注云计算服务在运行过程中，如何对安全事故、数据破坏或者其他严重的事件及时调查并采取相关的补救甚至司法行动。特别需要提到的是，与传统的安全事故响应流程相比，云计算服务的安全事故响应不同之处在于需要云服务提供商和用户的配合，双方具有共同的责任。

(4) 应用安全 (Application Security)，主要关注云计算平台上的应用安全问题。云计算用户利用云计算平台的能力 (无论是IaaS、PaaS还是SaaS) 编写自己的应用，这与传统的应用开发和运行有很大的不同，从而需要在应用的威胁模型和信任模型上做相应的改变。在云计算环境中，为了实现有效的应用安全，用户需要得到云计算服务提供商的支持。

(5) 加密和密钥管理 (Encryption and Key Management)，主要关注云环境下正确的加密和弹性的密钥管理。由于云计算环境的动态性、共享性和开放性 (尤其对于公共云而言)，通过加密对数据的私密性和完整性进行保护尤为必要。随着云计算业务的发展，云服务提供商也需要通过统一密钥管理方案以减轻用户使用的负担。

(6) 身份与访问管理 (IAM: Identity and Access Management)，主要关注用户身份的管理和通过相应的目录服务来提供访问控制。企业用户在将IT业务扩展到云计算业务时，需要考虑如何将已有的IAM系统扩展到云计算业务。比如，身份 (用户账号) 的发放和回收、强认证方式、认证联合、授权和用户概要 (User Profile) 管理等方面。如果云计算提供商以服务的方式提供了IAM，则用户还需要考虑企业内部的IAM与云计算提供商的IAM互相配合的问题。

(7) 虚拟化 (Virtualization)，此领域主要关注由于虚拟化在带来灵活性和动态性的同时所带来的一系列安全性问题，比如：多个企业共享同一套物理设备，物理设备的限制导致虚

拟机之间的相关性等。一般来说，云服务提供商具有更多的责任来保证各虚拟层次上的安全性。比如：划分不同的安全区域供用户选择，加强区域边界的安全措施，保证暴露在外的访问接口（如Web接口）的安全控制，提高虚拟机引擎（Hypervisor）的安全能力等。

总之，在保证云计算服务信息安全方面，关键的几点包括：

（1）应该通过合同、服务等级协议等方式明确提供商和用户在安全方面的责任和所采取的措施；

（2）应该通过提供商和用户之间的合作来构建立体的安全控制；

（3）应该保留相关的资料和证据以满足法律和审计的要求；

（4）应该关注最新的安全问题和应对措施，从而利用最新的安全控制方式。

由于云计算服务将大量的管理责任转移给了云计算提供商，提供商的安全措施是否完整和有效将是决定其是否被用户接受的关键因素之一。云计算提供商在保证安全方面具有更大的责任。

3.7 小结

云计算作为一种新的IT服务模式，有助于企业降低IT成本，提高服务水平。然而，正像任何一种技术进步所带给人们的一样，云计算对于企业来讲既是机遇也是挑战。深入理解这点对于企业达到预期的经济效益是至关重要的。在云计算的商业系统中，包括云计算的提供商和云计算的用户，他们或者构建云计算平台提供云计算服务，或者订阅并使用提供商的服务来实现自己的IT业务流程。

无论是云计算的提供商还是用户，在实施云计算时，首先需要对所拥有的整个IT基础设施做一个通盘考虑，制定一个明确的战略，并规划出切实可行的行动方案；然后在人力、财力、政策、商业环境允许的情况下，有计划按步骤地付诸实施。对于云计算的提供商而言，还应该有效管理并维护云计算服务平台，保证云计算服务的安全性、业务连续性、可靠性和使用的便捷性。云计算提供商和云计算用户之间密切合作才能够完整展现云计算服务的效益，从而有效支撑企业的成功。

第二篇 技术基石篇

第4章 新一代绿色数据中心

第5章 虚拟化概论

第6章 虚拟化管理

第 4 章

新一代绿色数据中心

- 4.1 数据中心概述
- 4.2 数据中心的设计和构建
- 4.3 数据中心的管理和维护
- 4.4 新一代数据中心的需求
- 4.5 绿色数据中心
- 4.6 小结



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的。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。

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。

4.1 数据中心概述

4.1.1 数据中心的概念

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。具体来说，数据中心是在一幢建筑物内，以特定的业务应用中的各类数据为核心，依托IT技术，按照统一的标准，建立数据处理、存储、传输、综合分析的一体化数据信息管理体系。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。

图4.1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、计算设备和业务信息系统这三个逻辑部分组成。支撑系统主要包括建筑、电力设备、环境调节设备、照明设备和监控设备，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。计算设备主要包括服务器、存储设备、网络设备、通信设备等，这些设施支撑着上层的业务信息系统。业务信息系统是为企业或公众提供特定信息服务的软件系统，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。



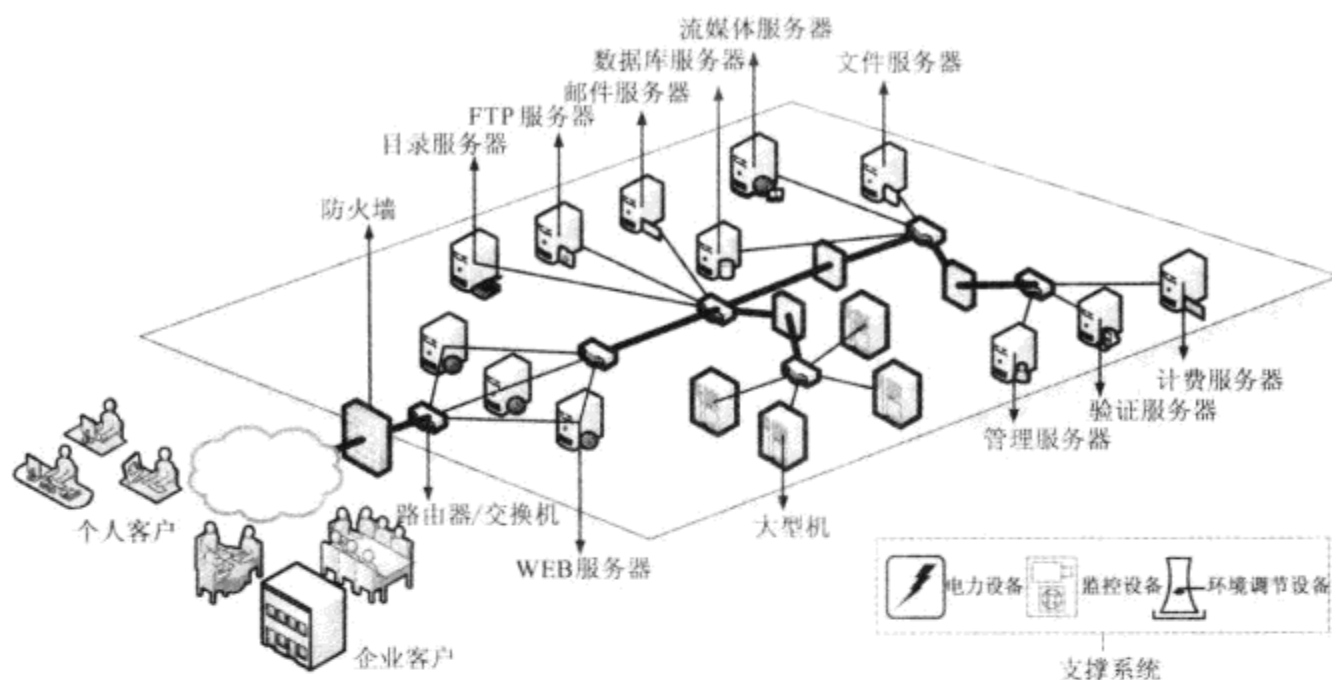


图4.1 数据中心逻辑示意图

可见，数据中心的概念既包括物理的范畴，也包括数据和应用的范畴。数据中心容纳了支撑业务系统运行的基础设施，为其中的所有业务系统提供运营环境，并具有一套完整的运行、维护体系以保证业务系统高效、稳定、不间断地运行。

4.1.2 数据中心的发展过程

早期的数据中心可以追溯到20世纪50年代，数据中心是存放大型主机的机房。当时的大型主机主要用于科学研究机构或国防军事领域，这些大型主机的主要器件以晶体管和电子管为主，占地面积大，价格也十分昂贵。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70年代以后，随着大规模集成电路的快速发展，计算机价格迅速下降，性能也飞速提升。发展到20世纪80年代，计算机向微型机的方向不断演进，只要购买一台廉价的个人计算机，即可完成很多计算任务。在这一阶段，计算机的发展由集中走向分布，小型机房得到了快速的发展。

进入到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。互联网将全球的计算机整合在一起，使得数据中心的发展又从分布逐渐走向了集中。互联网的蓬勃发展掀起

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。

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，数据中心的建设规模和服务器数量每年都在以惊人的速度增长。

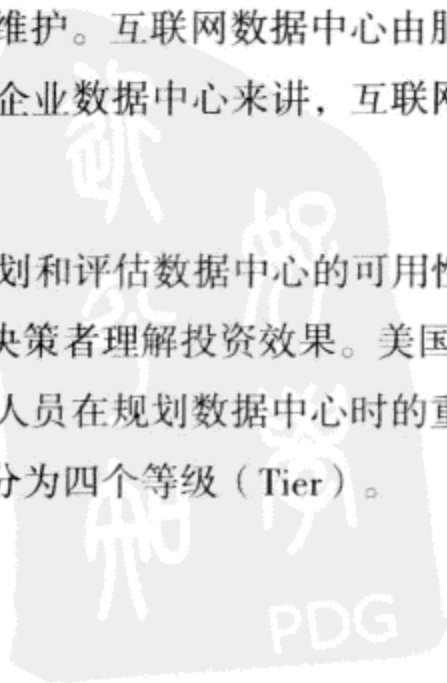
如今，飞速发展的信息服务和对IT系统的要求给数据中心带来了新的挑战。以往的企业数据中心往往只简单地追求计算力与性能，而当前的经济环境让企业更加注重数据中心的成本，绿色、节能、低碳的概念也逐渐深入人心。新一代的绿色数据中心通过自动化的管理方式、虚拟化的资源整合方式，结合新的能源管理技术，来解决数据中心日益突出的管理复杂、能耗严重、成本增加及信息安全等方面的挑战，实现高效、节能、环保、易于管理的数据中心。

4.1.3 数据中心的分类与分级

依据业务应用系统在规模类型，服务的对象，服务质量的要求等各方面的不同，数据中心的规模、配置也有很大的不同。

数据中心按照服务的对象来分，可以分为企业数据中心和互联网数据中心。企业数据中心指由企业或机构构建并所有，服务于企业或机构自身业务的数据中心，它为企业、客户及合作伙伴提供数据处理、数据访问等信息服务。企业数据中心的服务器可以自己购买，也可以从电信级机房中租用，运营维护的方式也很自由，既可以由企业内部的IT部门负责运营维护，也可外包给专业的IT公司运营维护。互联网数据中心由服务提供商所有，通过互联网向客户提供有偿信息服务。相对于企业数据中心来讲，互联网数据中心的服务对象更广，规模更大，设备与管理更为专业。

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。

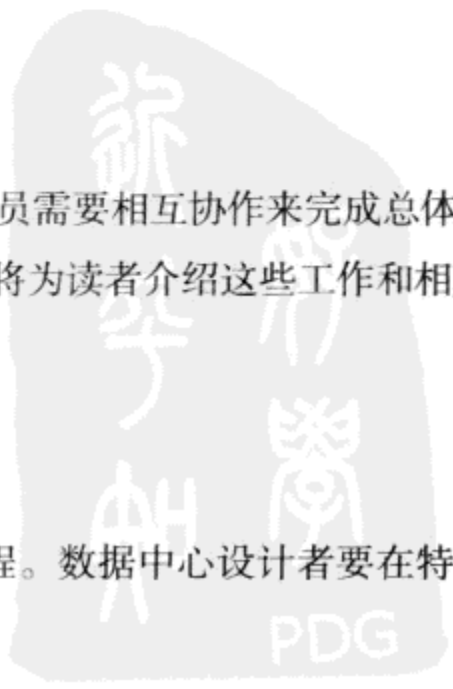
可见，随着等级的提高，数据中心具有了更强的可用性和整体性能。目前，已落成的数据中心在进行升级改造时都在力争达到Tier IV的要求。而面向云计算的下一代数据中心在设计时更是以Tier IV作为建设的标准。

4.2 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。

4.2.1 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，提高数据中心的效率和管理效率和整体性能。

建设数据中心的目的是为了满足不同企业信息化建设中的各项信息服务的需求，为它们提供高性能、高可用、高可扩展的安全的基础设施及软件平台。建设数据中心包括建设机房环境，为数据中心提供可靠、易用的电力、环境控制、消防等辅助配套设施，提供高效的网络环境，构建高效、稳定的服务器系统，存储系统，并建立安全体系和灾备系统。

构建数据中心需要遵守一些核心设计理念，遵守这些理念可以使得数据中心的设计清晰、高效、有条理。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系统机构和设备易于扩展，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；标准化的理念要求采用先进成熟的技术和设计规范，保证能够适应信息技术的发展趋势；经济性的理念要求选用性价比高的设备，系统可以方便地升级，充分利用原有投资。

4.2.2 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修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修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，即可向全球提供服务。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。数据中心可以小到一个房间，大到一层楼甚至是整幢楼房。数据中心的规模取决于企业的需求和预算，这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。从土建角度来讲，数据中心楼板的承重要求高于普通建筑，因为数据中心的服务器一般比较密集，大型机柜、网络设备的重量大于普通的家具和办公设备。因此在设计建筑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。

数据中心对楼层的高度也有要求，设计时需要计算铺设地板和安装吊顶以后的净高。因为一般数据中心都采用下进线方式，地板下要覆设走线槽和通风通道，所以地板净高至少需要30~50厘米；而房顶吊顶中要留足灯具和消防设备暗埋高度，这样房间的净高至少累计减少70~80厘米，普通楼房的高度在机房装修后会显得较低，也不利于设备的安装。所以数据中心的房间净高度最好在3.3米以上。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。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，要在屋顶布置照明、防火、安全监控等设施。

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。

4.2.3 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4.2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。

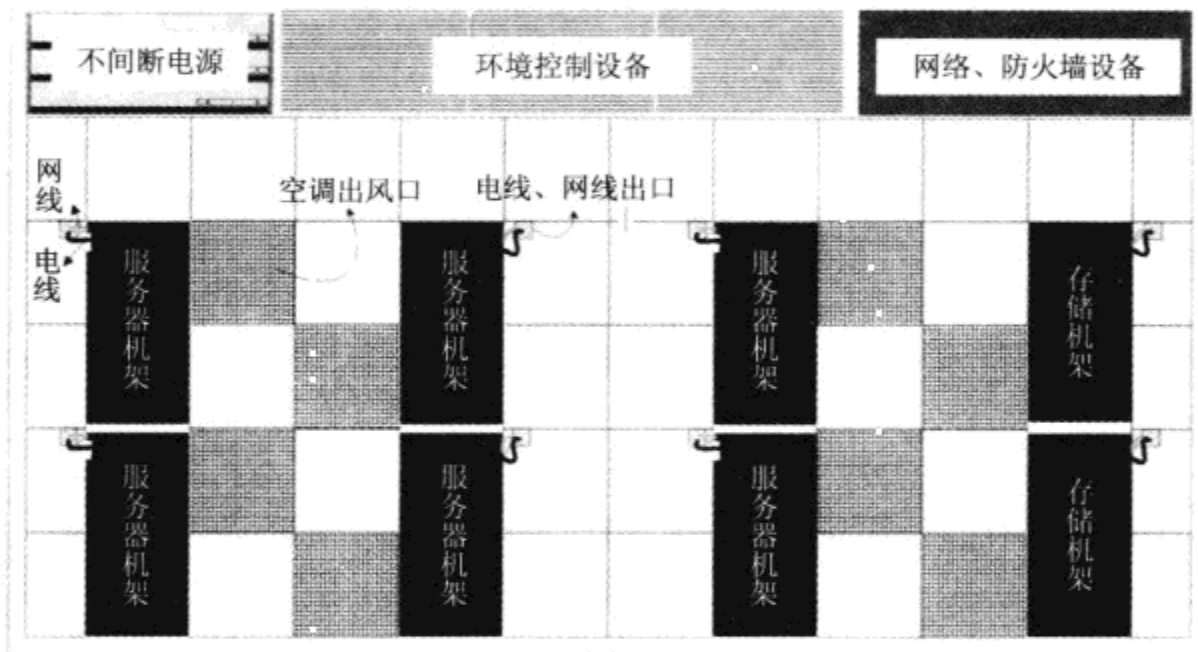


图4.2 数据中心基础设施示意图

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、稳定地运行。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。

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4.3所示，为了布线方便，一般都将机房地面架空，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。

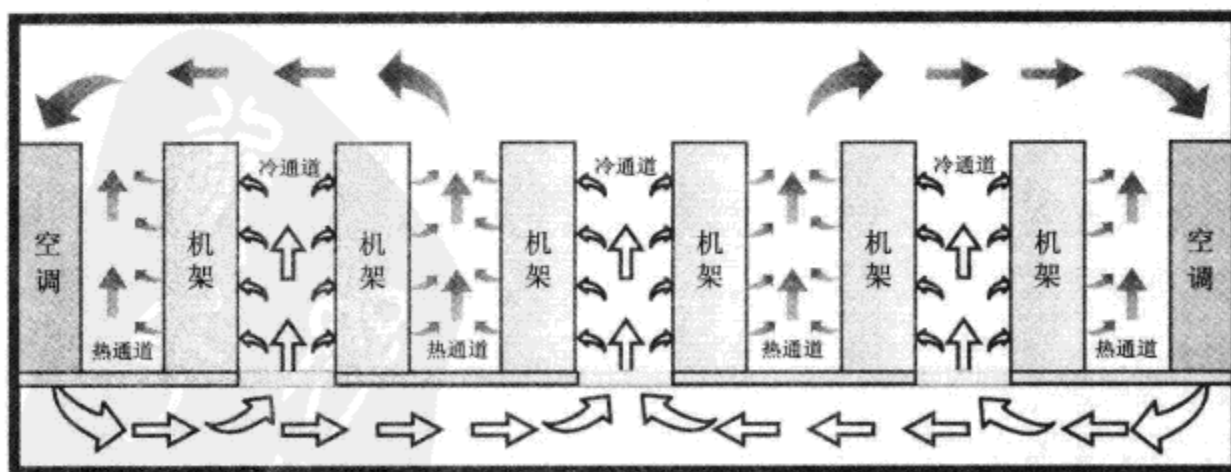


图4.3 数据中心风冷示意图

如果数据是血液，网络就是血管。网络系统是信息的高速公路，在数据中心内及数据

中心之间起着至关重要的作用。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。

数据中心的网络设备主要有交换机和路由器。交换机是一种基于MAC地址识别的封装转发数据包功能的网络设备。与集线器共享带宽的广播方式不同，交换机可以识别数据帧的发送者和目标接收者，使数据帧直接从源地址到达目的地址。通过交换机的过滤和转发，可以有效地解决广播风暴问题，减少误包和错包的出现，避免共享冲突。

从传输速度上来分，局域网交换机可以分为以太网交换机、快速以太网交换机、千兆以太网交换机等。插槽与扩展槽数、支持网络类型、背板吞吐量、最大可堆叠数等是选取以太网交换机的主要参数。路由器用于连接多个逻辑上分开的网络，当收到数据时，通过路由规则判断网络地址并选择路径，完成数据在多个子网间的传输。路由器的主要工作目标是尽可能选择通常快捷的网络路径，提高通信速度，减轻网络系统通信负荷，节约网络资源，提高网络畅通率，从而让网络发挥出更大的效益。选购路由器时，应重点考虑路由器所支持的路由协议类型、吞吐量、转发延时、路由表的容量，以及路由器的稳定性等因素。

4.2.4 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。

塔式服务器的外观与个人计算机的主机差不多，如图4.4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，不便于挪动和管理。

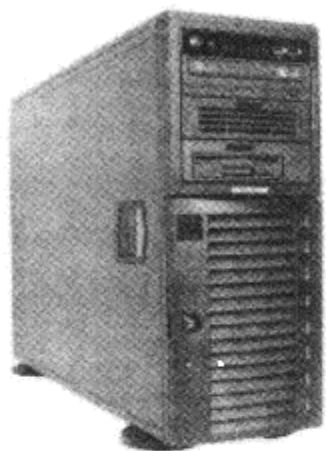


图4.4 塔式服务器

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4.5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4.5描绘了从1U到4U四种不同尺寸的机架式服务器。与塔式服务器相比，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。

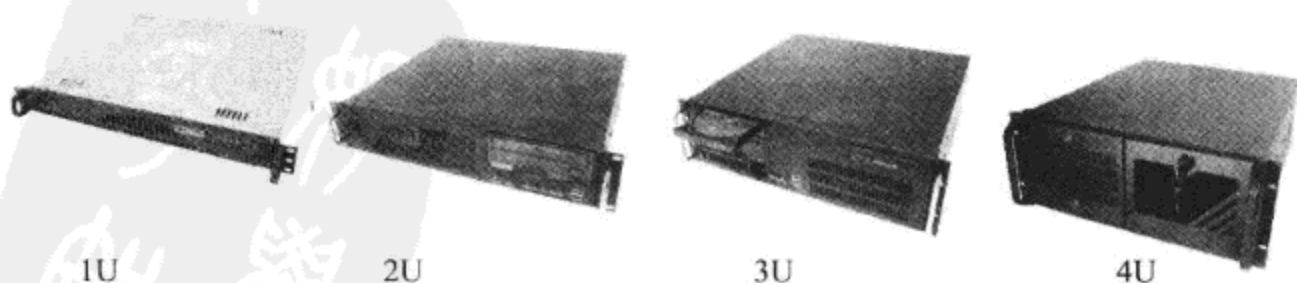


图4.5 机架式服务器

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4.6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，在一定程度上降低了成本。刀片服务器一般应用于大型数据中心或者计算密集的行业，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。

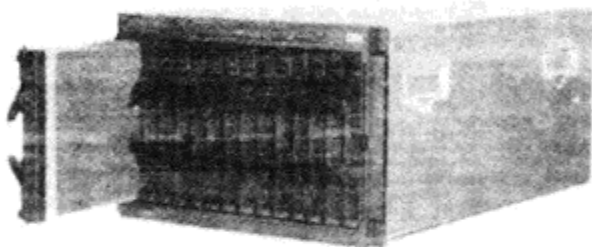


图4.6 刀片服务器

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要，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、Solaris等。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。

4.3 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。

4.3.1 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，定期清理除尘是必不可少的。

4.3.2 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。

4.3.3 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，并将在2010年突破300GB，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。

4.3.4 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。

(1) 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。

(2) 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。

(3) 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。

(4) 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。

4.3.5 安全管理

作为企业信息系统的核心，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。

(1) 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。

(2) 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，识别出非正常使用情况并加以封禁。

(3) 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。

(4) 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。

(5) 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、在系统中安装了哪些木马程序，以便快速定位和解决问题。

4.4 新一代数据中心的的需求

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、可靠性这几个方面分别讨论。

4.4.1 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。

4.4.2 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。

4.4.3 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、数据库软件的性能配置、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。

4.4.4 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很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。

4.4.5 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。

4.5 绿色数据中心

4.5.1 经济型数据中心

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4.7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。

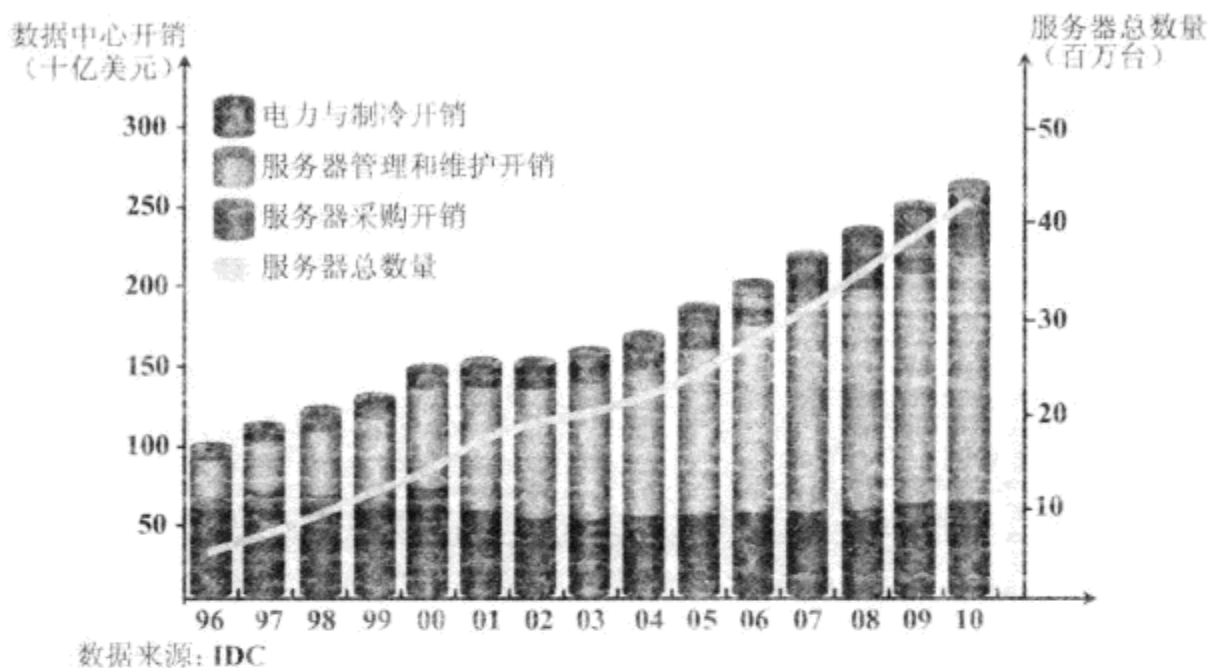


图4.7 数据中心成本构成及发展趋势

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设备更新换代很快，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。

4.5.2 数据中心能效分析

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。

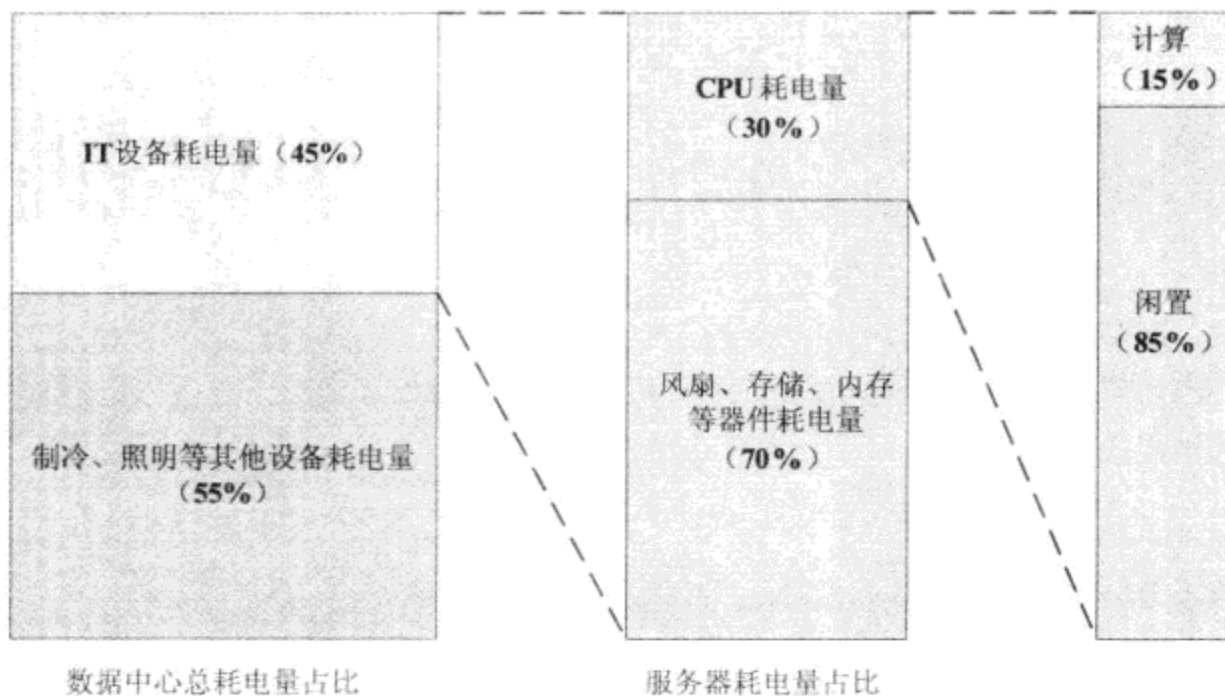


图4.8 数据中心耗电比例示意图

图4.8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。

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。

$$PUE = \frac{\text{总能耗}}{\text{IT设备能耗}}$$

从PUE的定义可以看出，PUE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。从这个角度来看，要想降低数据中心的总能耗，需要从降低IT设备能耗和降低PUE值两方面入手。

首先来谈谈如何降低IT设备的能耗。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。刀片服务器是当前数据中心服务器发展的主流趋势。刀片服务器体积小，通常一个刀片服务器机架可以插入8~16个刀片服务器，这些刀片服务器共用一个系统背板和冗余电源、风扇、网络端口和其他外部设备，通过这种共享方式，单个刀片服

服务器的能耗大大降低，服务器电源的工作效率也得到提升。因此，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。

如果将PUE公式中的总能耗分解，可得

$$\begin{aligned} \text{PUE} &= \frac{\text{总能耗}}{\text{IT设备能耗}} \\ &= \frac{\text{IT设备能耗} + \text{制冷设备能耗} + \text{供配电损耗} + \text{辅助系统能耗}}{\text{IT设备能耗}} \\ &= 1 + \text{制冷能耗因子} + \text{供配电能耗因子} + \text{辅助系统能耗因子} \end{aligned}$$

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

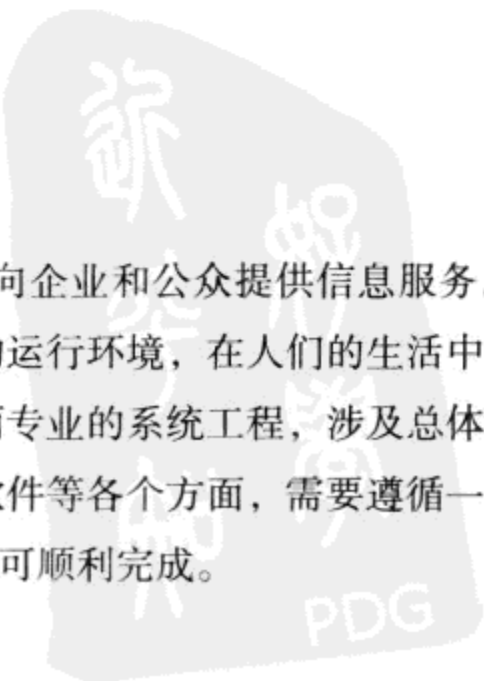
设施的设计和建设水平及所处环境的气候条件。据统计，国外先进的数据中心PUE值可达1.6~1.8，而国内的数据中心PUE值平均在2.0~2.5之间，中小规模的数据中心PUE值更高，有的甚至在3.0以上。近几年新设计建造的数据中心，PUE值可以达到1.8左右。一个典型的PUE为2.2的数据中心，IT设备耗电量占45%，空调设备耗电量占45%，而照明等其他设备耗电量之和不过10%，可以看出，一半以上的电量都被空调等设备消耗了。

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。

4.6 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。



第 5 章

虚拟化概论

- 5.1 虚拟化的定义
- 5.2 服务器虚拟化
- 5.3 其他虚拟化技术
- 5.4 小结



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，并讨论在数据中心被广泛采纳的其他虚拟化技术。

5.1 虚拟化的定义

5.1.1 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。

5.1.2 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：

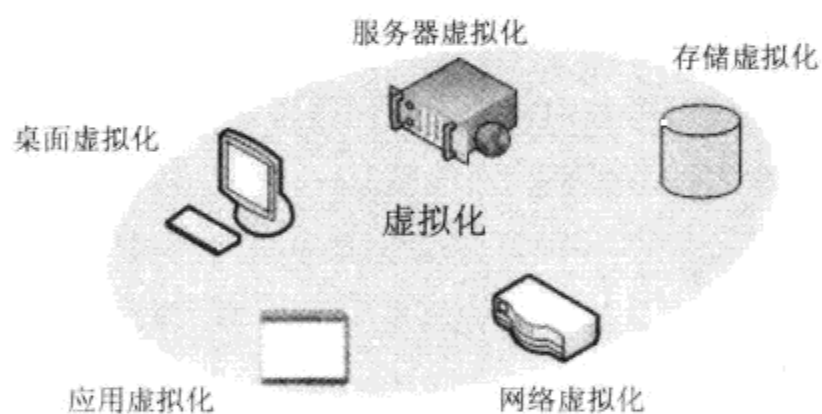
- ▶ 虚拟化的对象是各种各样的资源；
- ▶ 经过虚拟化后的逻辑资源对用户隐藏了不必要的细节；

► 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5.1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。



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。

5.1.3 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。

5.1.3.1 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。

文件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—个控制台上管理分散在不同位置的异构设备上的数据。

5.1.3.2 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11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5.2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。

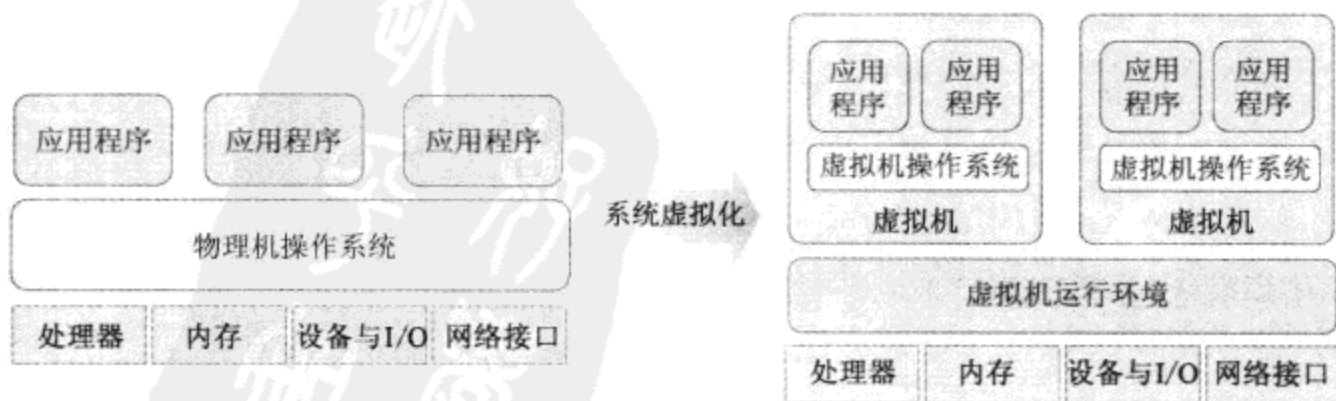


图5.2 系统虚拟化

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往往托管了数以万计的x86服务器。出于安全、可靠和性能考虑，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。

除了个人电脑和服务器的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。

5.1.3.3 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5.2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，在交付应用时，软件虚拟化也会为数据中心的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5.3节介绍。

5.2 服务器虚拟化

5.2.1 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，在一个物理服务器上创建出若干个可独立使用的虚拟机服务器。如图5.3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。

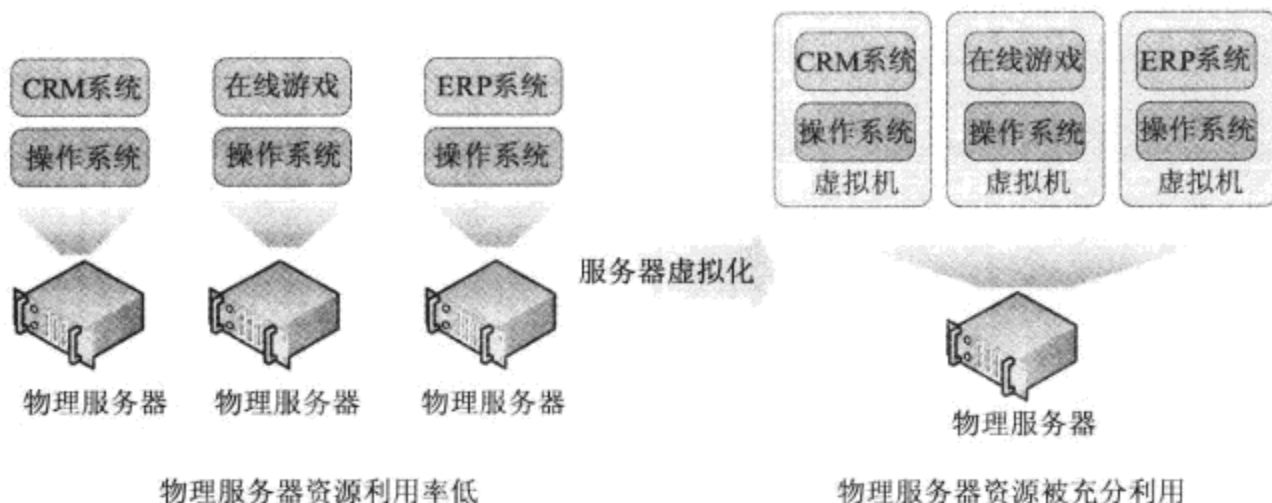


图5.3 服务器虚拟化

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11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。

5.2.2 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，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5.4所示。表5.1比较了这两种实现方式。

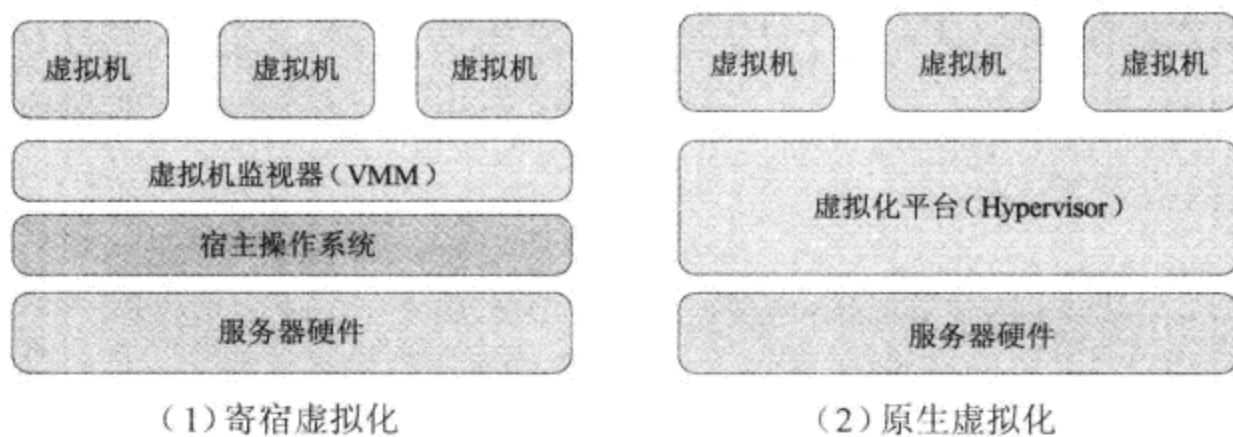


图5.4 服务器虚拟化的实现方式

表5.1 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。

5.2.3 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

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。

5.2.4 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。

5.2.4.1 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Ring 0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5.5所示。

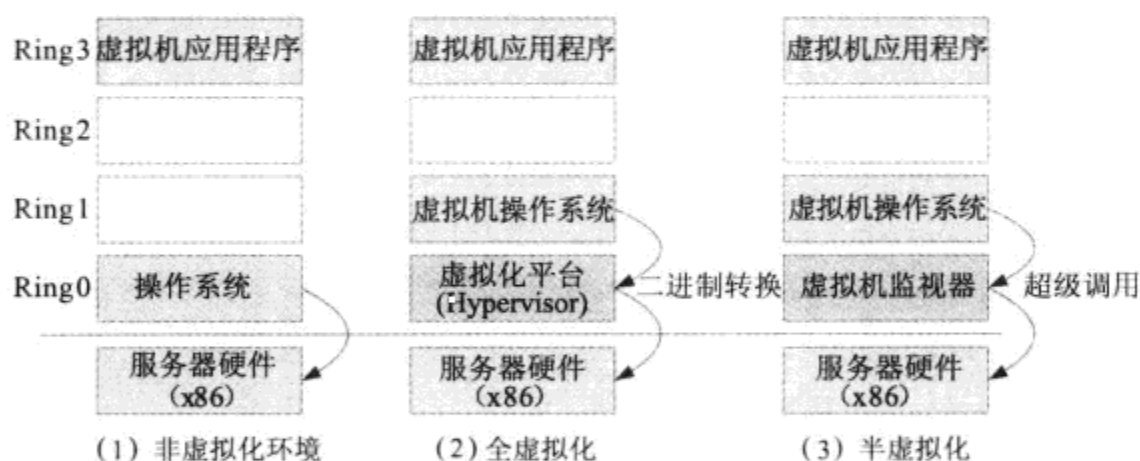


图5.5 x86体系结构下的软件CPU虚拟化

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5.5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5.5（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，无须修改客户操作系统，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5.5（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，无须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。

5.2.4.2 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5.6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。

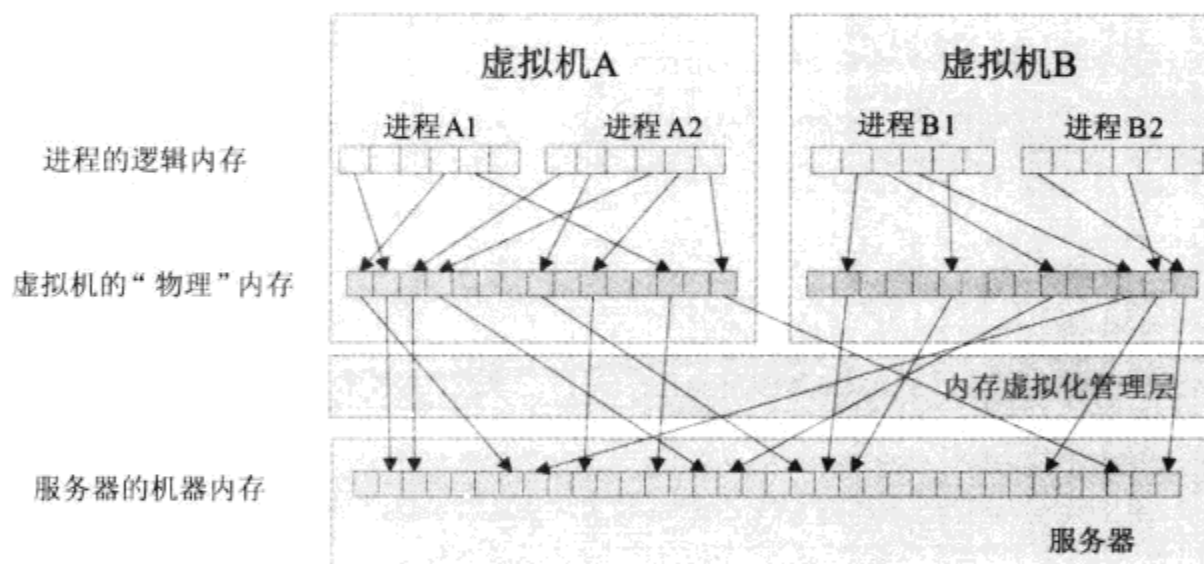


图5.6 内存虚拟化

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5.7（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5.7（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。

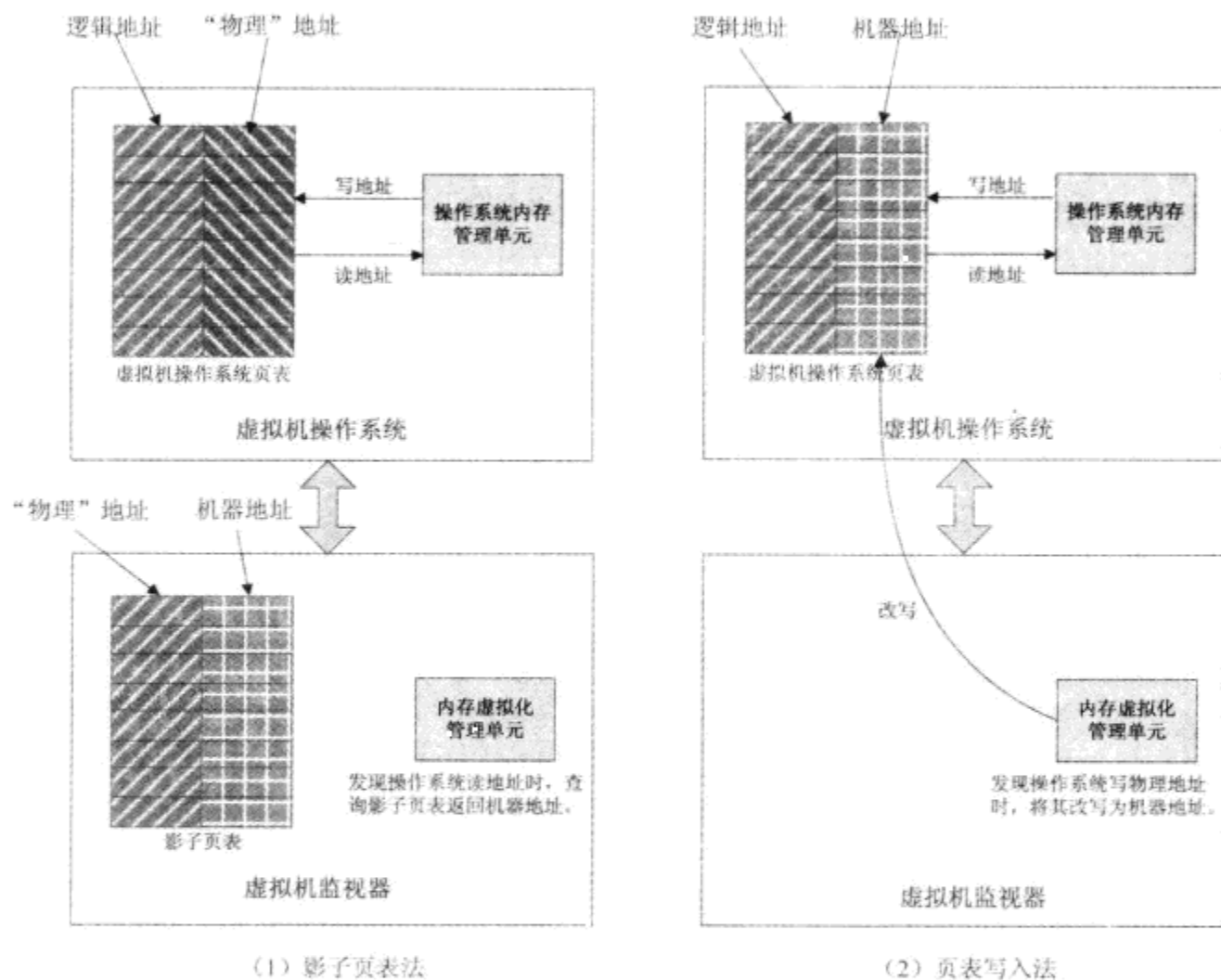


图5.7 内存虚拟化的两种方法

5.2.4.3 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5.8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。

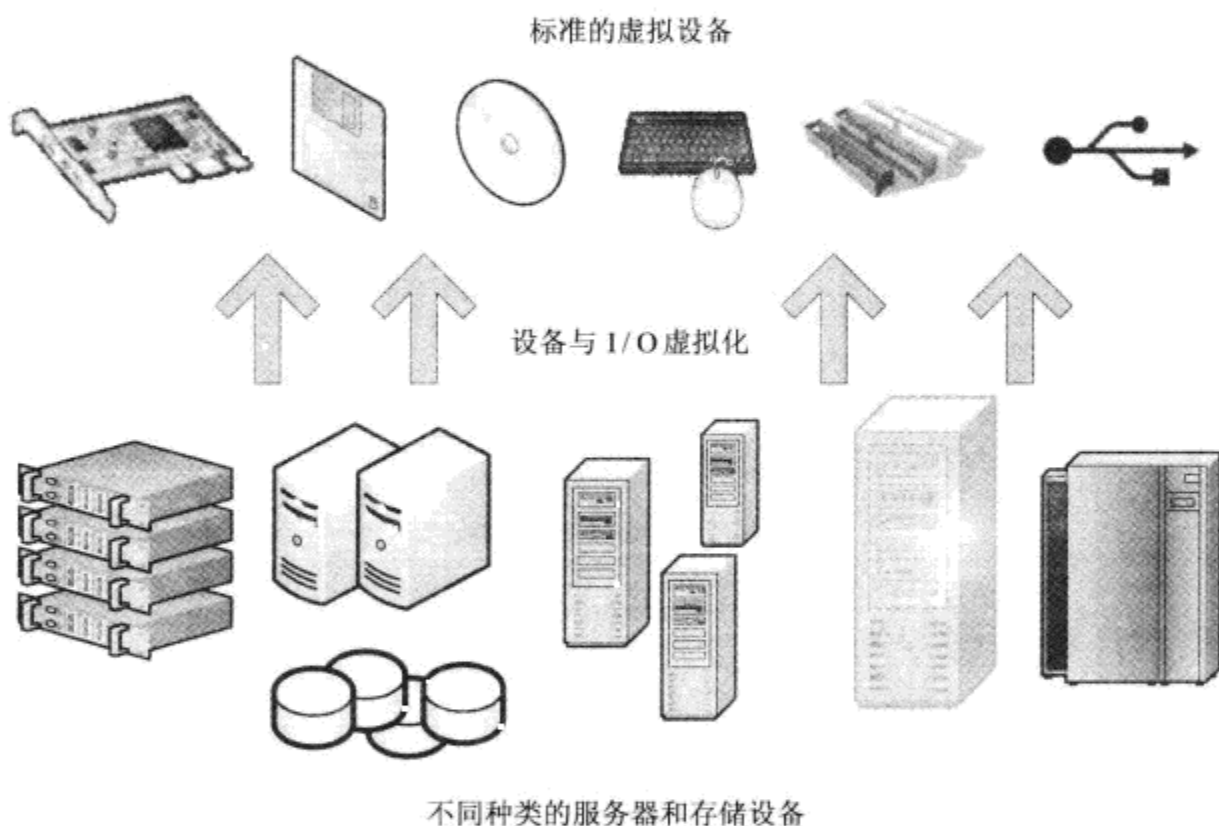


图5.8 设备与I/O虚拟化

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5.9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。

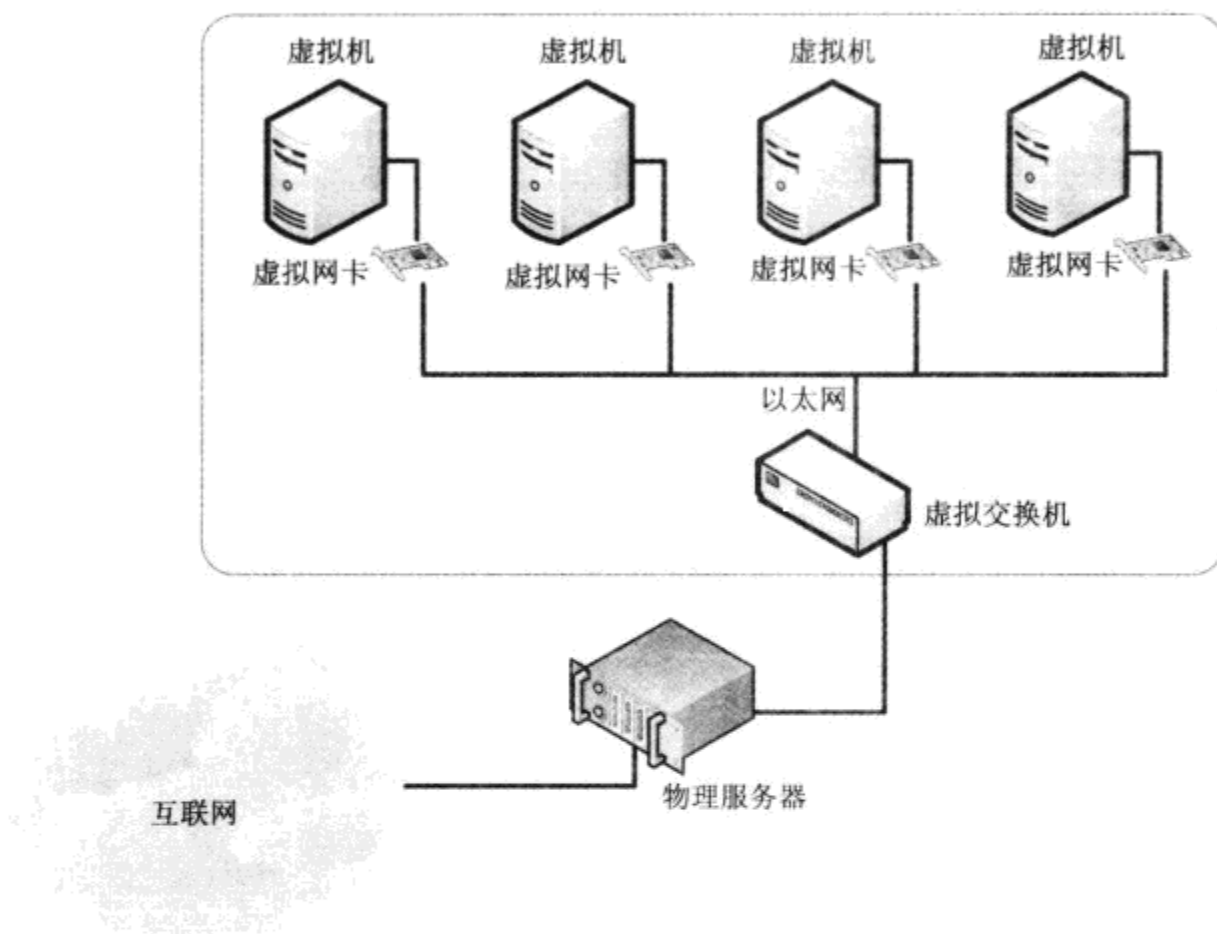


图5.9 网络接口虚拟化

5.2.4.4 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5.10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。

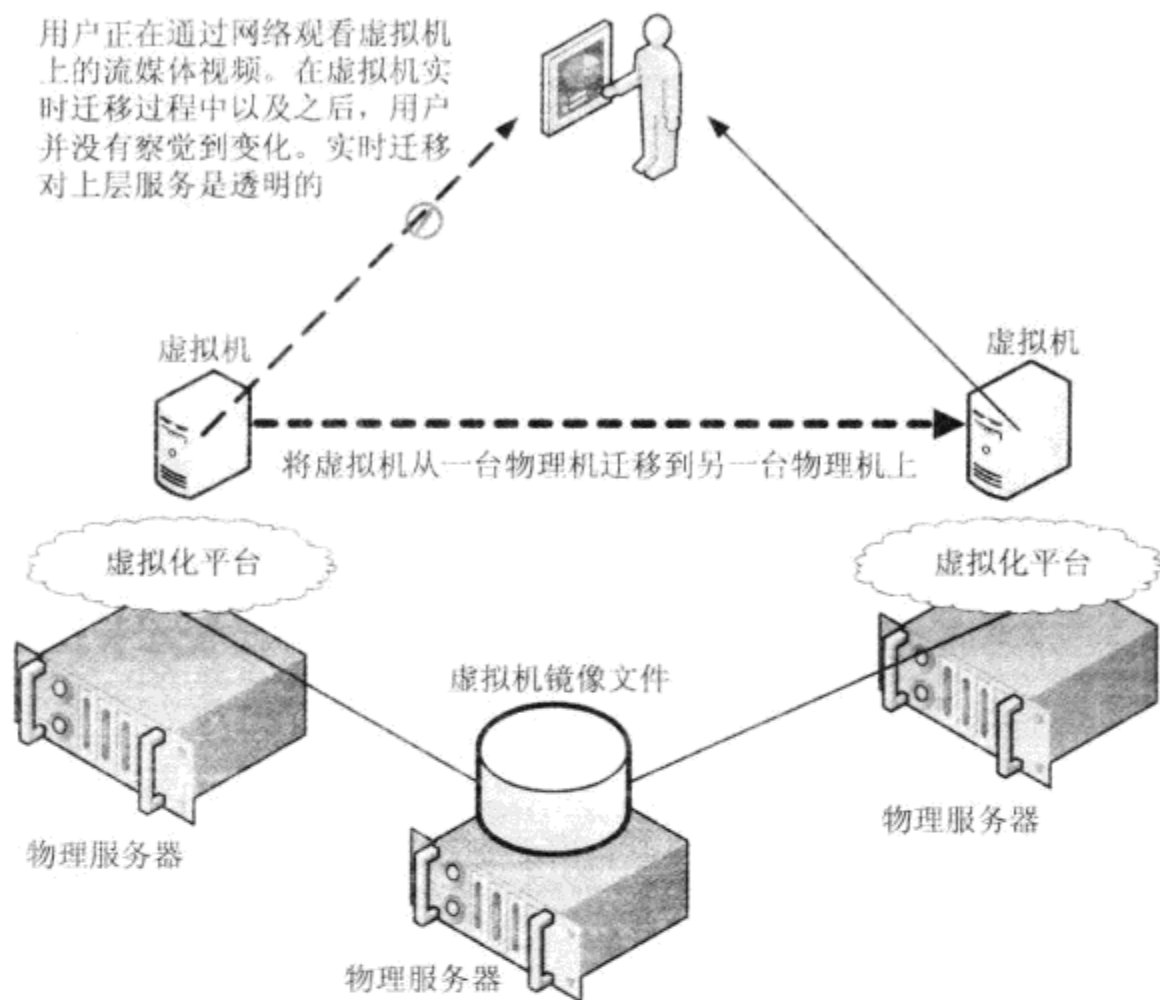


图5.10 实时迁移技术示意图

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。

5.2.5 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 Enterprise 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。

5.2.6 技术优势

5.2.6.1 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。

5.2.6.2 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。

5.2.6.3 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。

5.2.6.4 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。

5.2.6.5 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的，出于管理简便、安全性和性能考虑，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。

5.2.6.6 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。

5.2.6.7 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。

5.3 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應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的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。

5.3.1 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。

5.3.2 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的，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。

5.3.3 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5.11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。

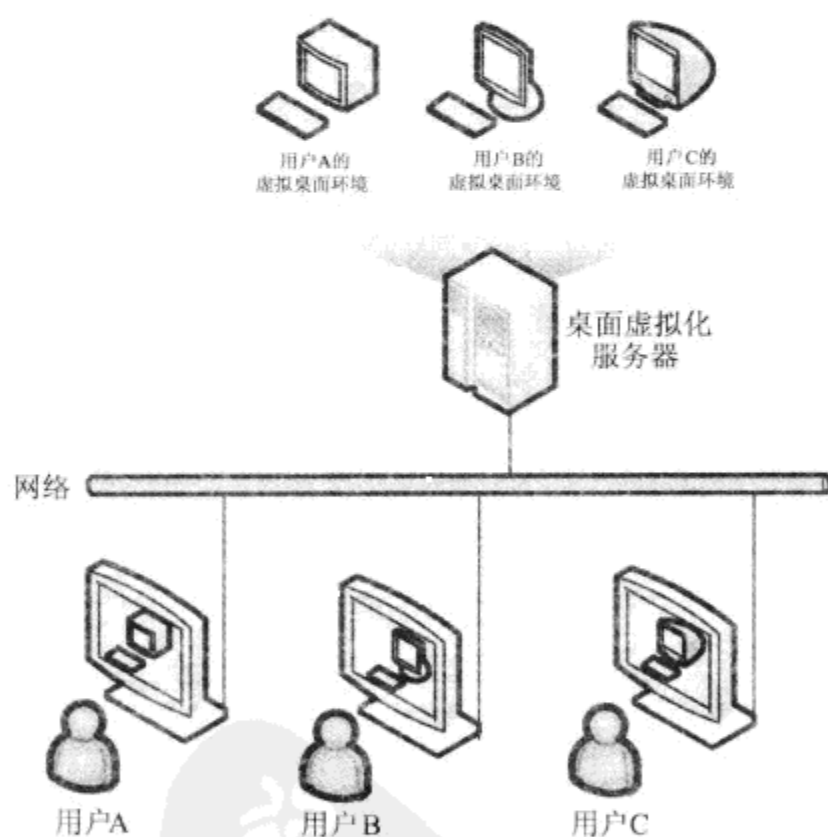


图5.11 桌面虚拟化

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得自身相关数据，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。

5.3.4 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5.12所示。

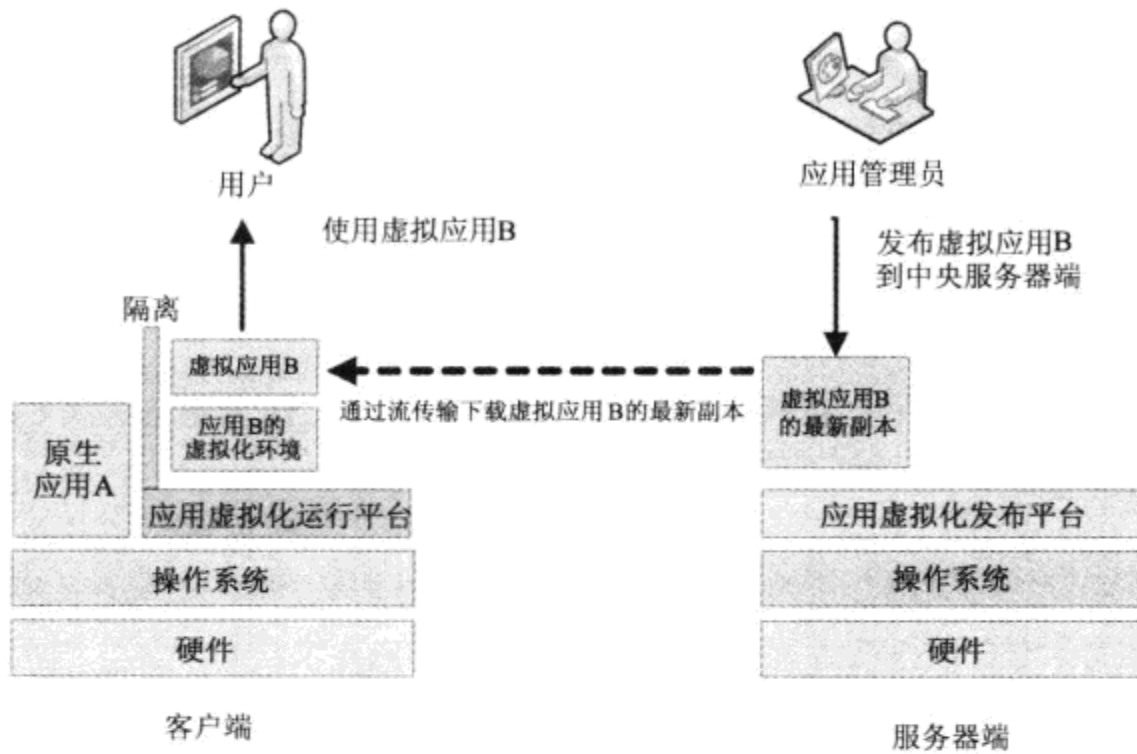


图5.12 应用虚拟化

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的，而是从一个中央服务器上下载下来的，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。

5.4 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的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6章，我们将为读者介绍实施服务器虚拟化的关键技术。在第11章，我们将概述虚拟化技术的最新业界动态。



第 6 章

虚拟化管理

- 6.1 创建虚拟化解决方案
- 6.2 部署虚拟化解决方案
- 6.3 管理虚拟化解决方案
- 6.4 小结



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6.1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。

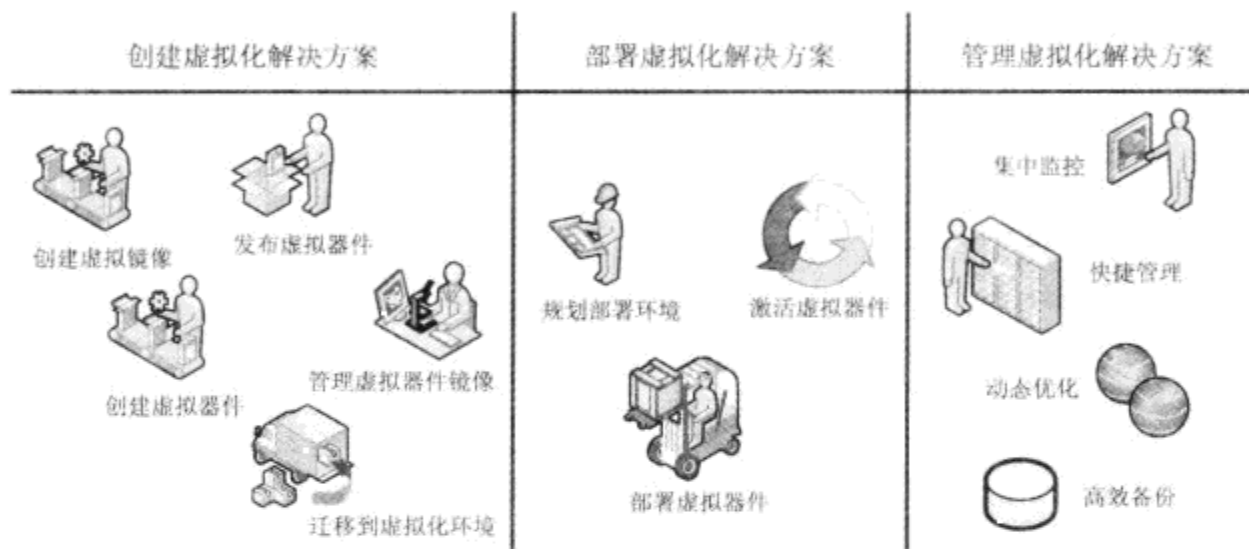


图6.1 虚拟化解决方案生命周期示意图

6.1 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。

6.1.1 创建基本虚拟镜像

根据第5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6.2.2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6.2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。

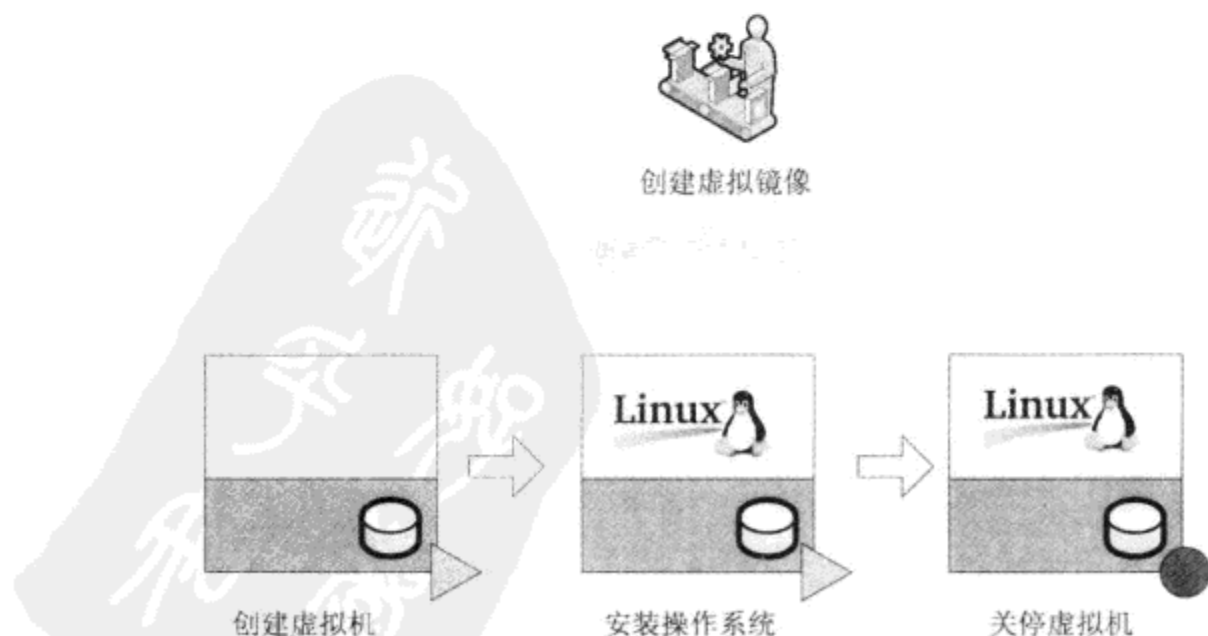


图6.2 创建虚拟镜像流程图

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。

6.1.2 创建虚拟器件镜像

在上一节中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6.3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。

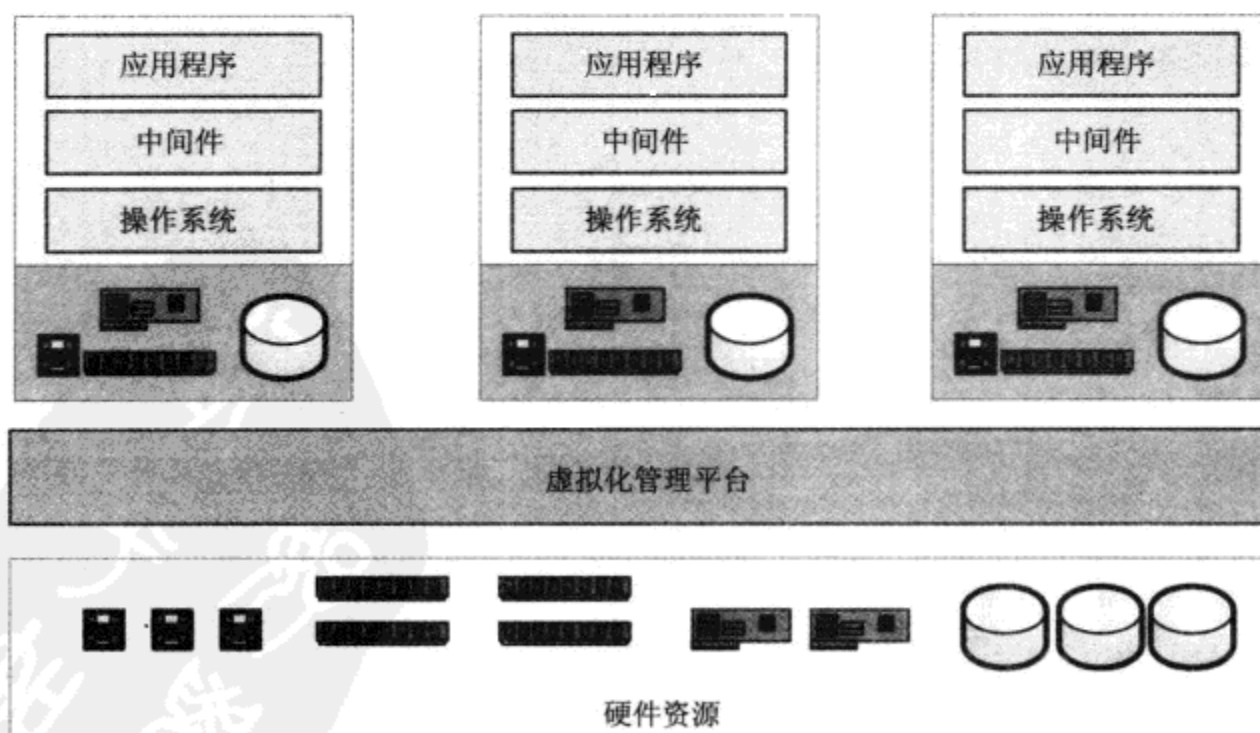


图6.3 虚拟器件结构图

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6.4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。

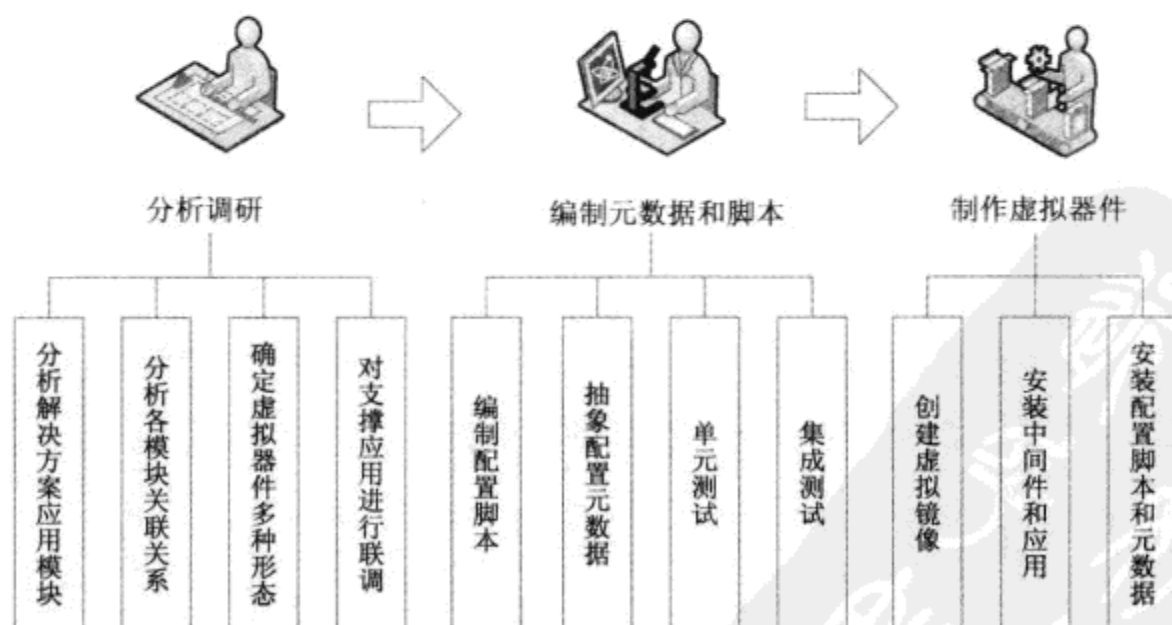


图6.4 虚拟器件创建流程图

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6.5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6.6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。

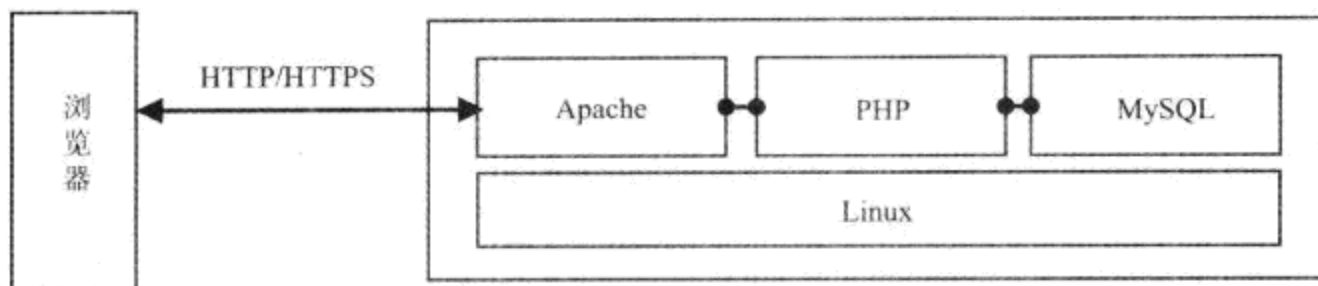


图6.5 LAMP 解决方案

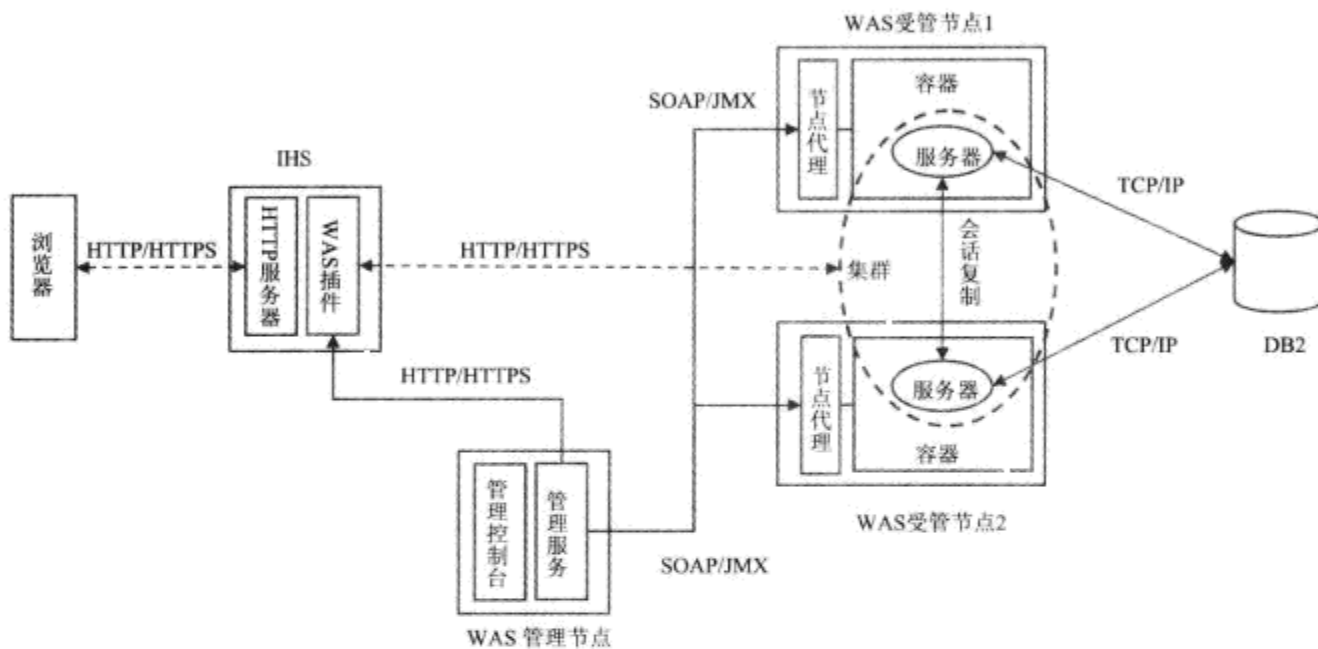


图6.6 IHS-WAS-DB2 解决方案

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序执行。

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。

6.1.3 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6.2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OVF1.0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6.7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。

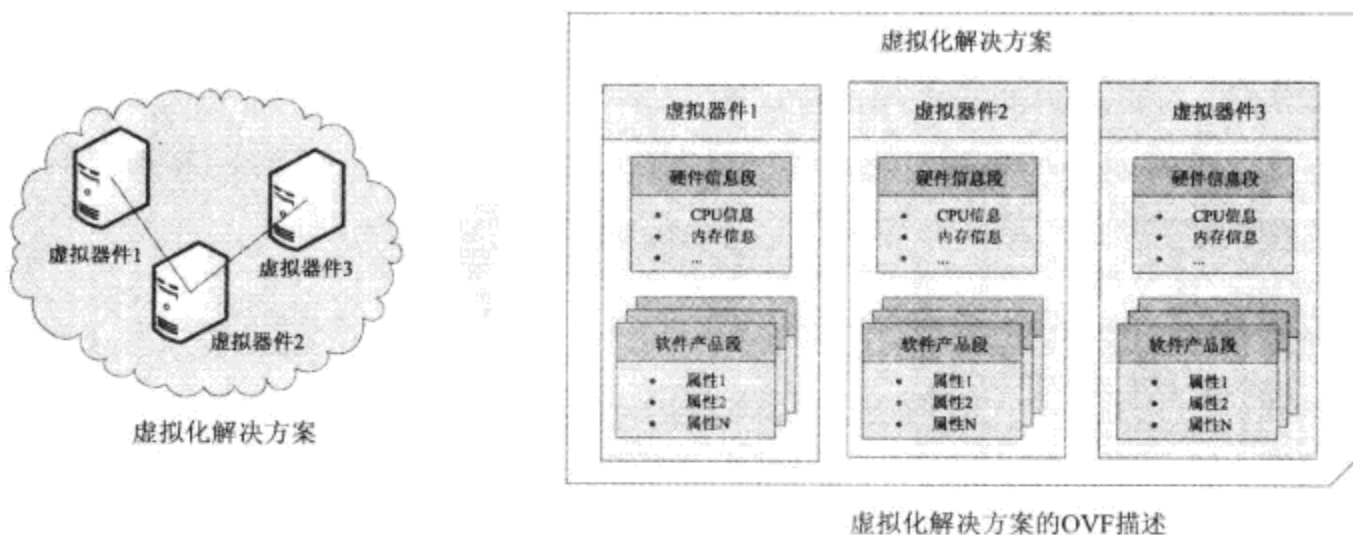


图6.7 OVF描述文件结构示意图

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6.8所示。

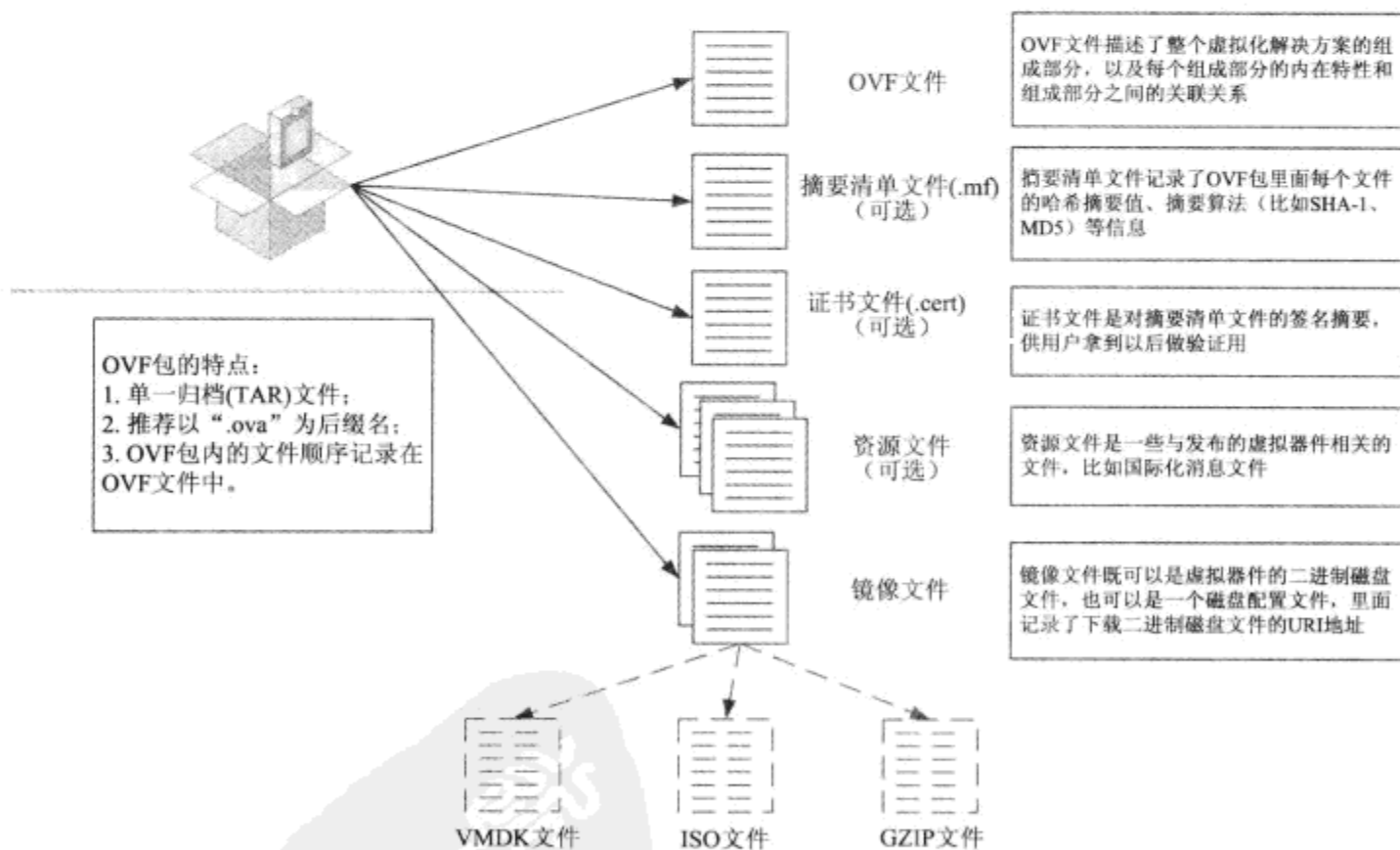


图6.8 OVF包结构示意图

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需的。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。

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。

6.1.4 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。

6.1.5 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6.9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。

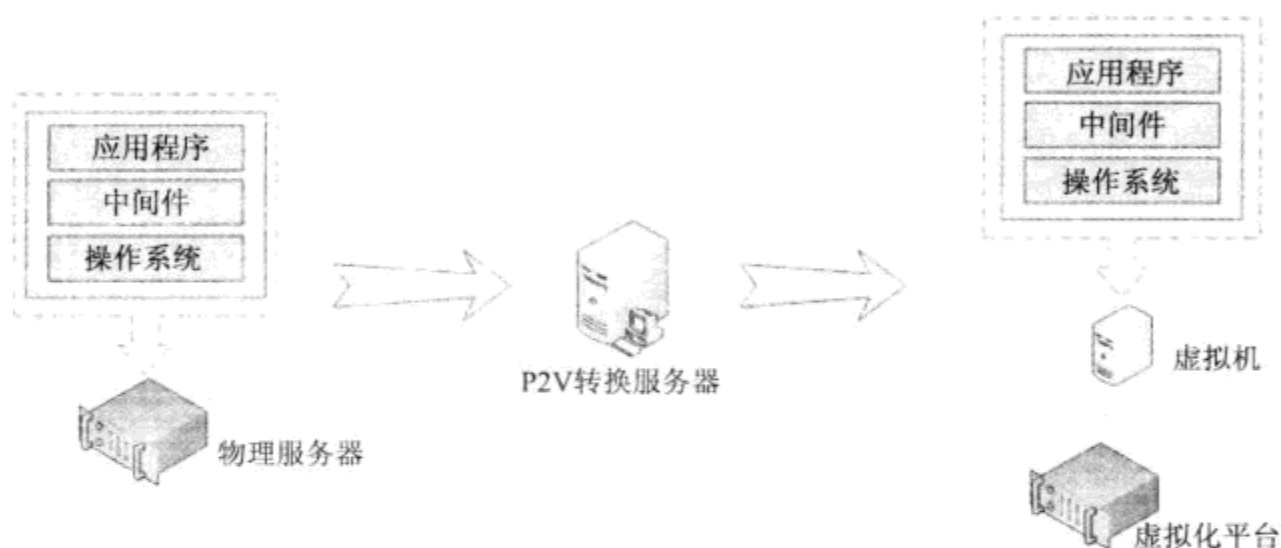


图6.9 P2V示意图

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。

6.2 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6.2.1节“规划部署环境”中讲述。6.2.2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6.2.3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。

6.2.1 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2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 Server2008系统一同发售的，而Xen有两个版本：商业版（Citrix Xen Server）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。

6.2.2 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6.1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6.10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6.2.3节中单独介绍。

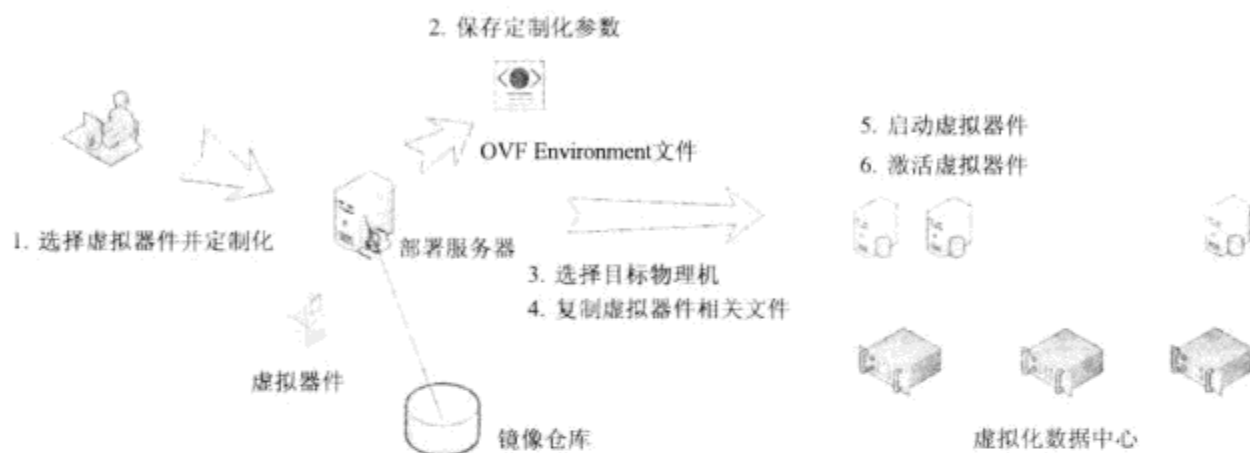


图6.10 部署虚拟器件流程图

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6.1.3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，另外一个部分是传输部分。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，再将它们与第2步生成的OVF Environment文件、虚拟机配置文件一起复制到目标物理机上。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须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6.1.4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，镜像管理系统无须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复制到虚拟化平台，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有一个磁盘设备，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。

6.2.3 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。

6.1.2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 6.1.2 节中提到，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。

6.3 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的调度，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。

6.3.1 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。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6.11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。

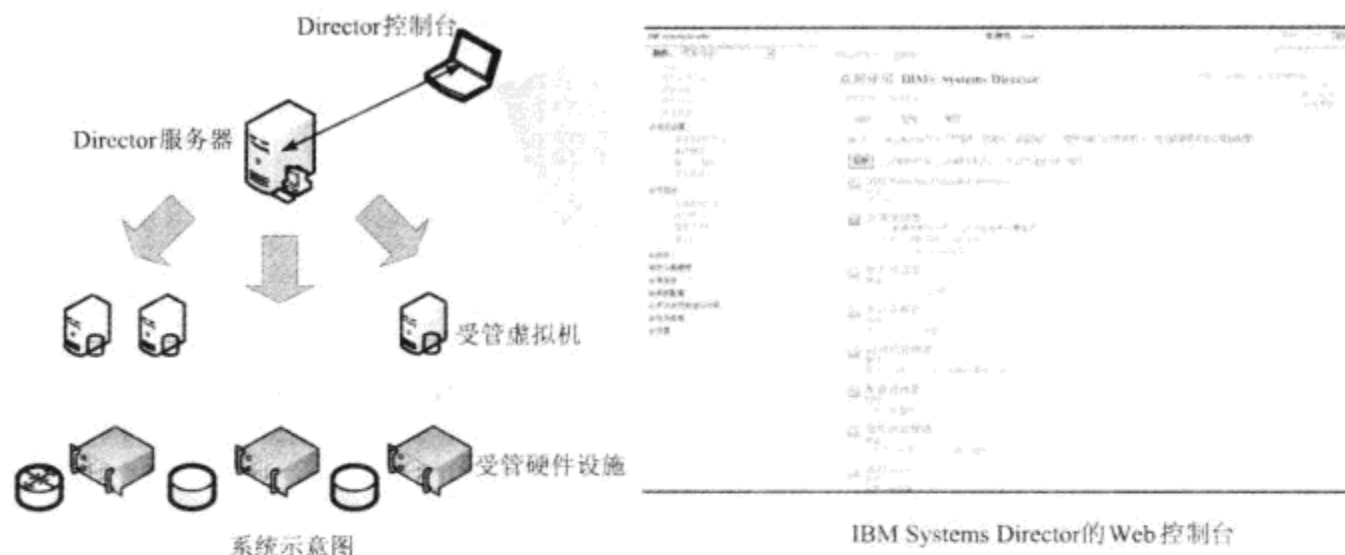


图6.11 IBM Systems Director集中控制台

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数据中心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。

6.3.2 快捷管理

在数据中心的，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6.1.2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现有技术，因而虚拟机简化操作成为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，而无须关心下层虚拟化平台的特殊性。

6.3.3 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它们的资源状态，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，但本质上讲它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，实现预防性的调整；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须移动镜像，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，期间无须停止任何应用程序，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。

6.3.4 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。

（1）大量具备高度相似的内容的虚拟机镜像并存。在传统的状况下，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。

（2）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。

（3）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。

（4）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。

(5) 虚拟机可以实时迁移，从文件系统的备份角度来讲，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。

(1) 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同样的备份代理软件，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快速备份，而且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

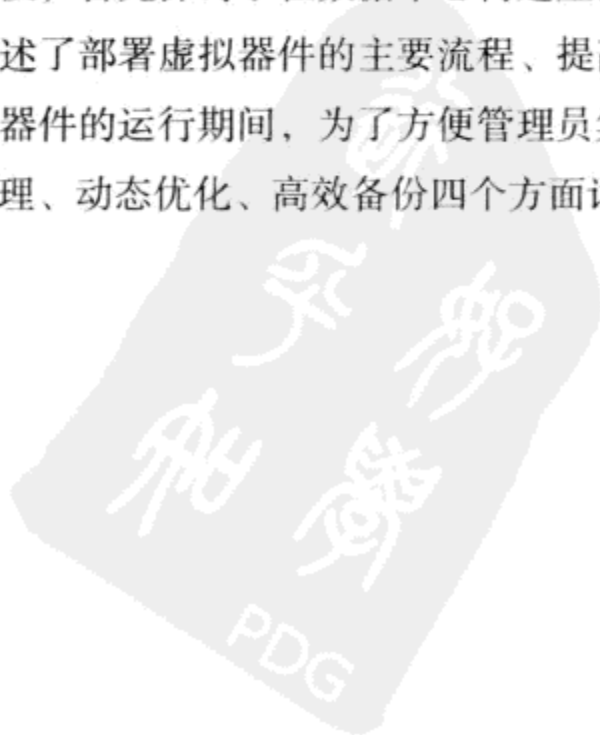
余数据删除，仅存档一份数据。

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。

6.4 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。



第三篇

系统架构篇

- 第7章 云架构
- 第8章 基础设施即服务
- 第9章 平台即服务
- 第10章 软件即服务

「第三篇」
系统架构篇



第 7 章

云架构

- 7.1 云架构的层次
- 7.2 云架构的特性
- 7.3 云架构的准则
- 7.4 小结



作为一种新兴的计算模式，云计算能够将各种各样的IT资源、应用和平台等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、运行应用的中间件平台及虚拟的服务器、存储等IT基础资源。当云计算环境将硬件、软件和各种相关技术有效地组织在一起并作为云计算服务提供时，我们应当考虑需要面临的大规模性、可用性、可伸缩性及其对性能、信息安全等方面的要求。云计算系统的组织方式即为云计算架构（云架构）。本章将概括介绍典型云架构的基本层次及各个层次的主要功能，作为后续三个章节的引子。

7.1 云架构的层次

7.1.1 云架构的基本层次

我们知道，从基本的功能来看，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。如图7.1所示，由美国国家标准研究院（NIST）提出，后被广泛引用的云架构包含三个基本层次：基础设施层（Infrastructure Layer）、平台层（Platform Layer）和应用层（Application Layer）。该架构层次中每层的功能都以服务的形式提供，这就是云服务类型分类方式的来源，即从云架构不同层次所提供的服务来进行划分。

基础设施层以IT资源为中心，包括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，并且实现了高效管理、操作流程自动化和资源优化，从而向用户提供动态、灵活的基础设施层服务。

平台层以平台服务和中间件为中心，介于基础设施层和应用层之间。平台层包括具有通用性和可复用的软件资源的集合，是优化的“云中间件”，提供了应用开发、部署、运行相关的中间件和基础服务，能够更好地满足云应用在可伸缩性、可用性和安全性等方面的要求。

应用层是云上应用程序的集合，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，也可以是定制的客户应用，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；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，一般部署在公有云平台上，比如金融行业需要的台账系统，电信行业需要的工单管理系统等。

需要注意的是，某个云计算提供商所提供的云计算服务可能专注在云架构的某一层，而无须同时提供三个层次上的服务。如第1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。

图7.1所示为逐层依赖的云架构。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；Salesforce CRM不仅要提供对底层硬件和中间软件平台的支持，还要为用户开发立即可用的软件或软件功能模块。

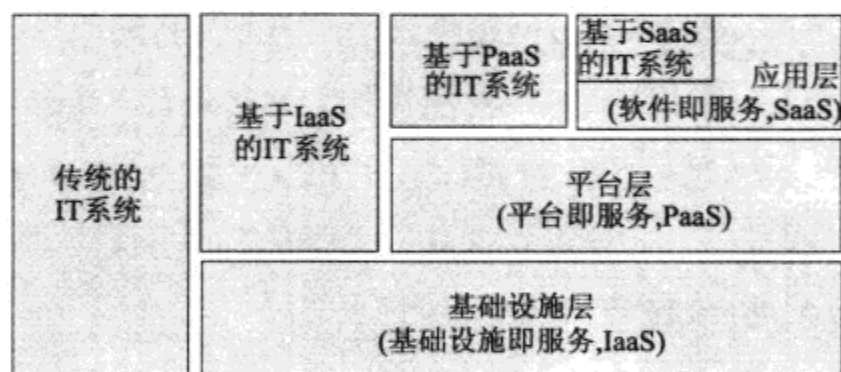


图7.1 云架构层次示意图

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。事实上，上层服务的提供者可以利用那些位于下层的云计算服务来实现自己的云计算服务，而无须自己实现所有下层的架构和功能。可以预见，随着云计算服务的发展，位于不同架构层次的云计算提供商之间的联合和服务集成，将是开展云计算业务的重要方式。

图7.1从横向的角度展示了在云计算时代企业IT系统可能的实现方式。从左到右有四

种方式：首先是传统的IT系统，企业可以继续自建自营从硬件到软件到应用的整个IT系统栈，这也是传统IT系统的建设管理方式；其次，如果企业已有的软件系统是其所特定的，但是可以运行在由基础设施层提供的计算资源上，则企业可以将其软件系统转移到IaaS服务上去，从而减轻运营维护IT硬件的负担；更进一步，如果企业的应用系统运行在通用的软件平台上，它可以进一步采用云平台层所提供的PaaS服务，这样可以更大程度上减轻运营管理IT系统的负担；最为极致的程度就是企业直接采用云应用，自己不再拥有IT系统，而直接通过云服务来满足自己所需的各种软件服务。在云计算时代，企业采用何种IT系统形式取决于企业的IT战略发展规划，即IT系统的现状、发展方向和业务需要。总体来说，云计算因其带来的种种优势成为企业IT系统发展的未来趋势。

7.1.2 云架构的服务层次

图7.1也显示出云架构中的每一层都可以独立为用户提供服务，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。本节将介绍这些服务，使读者从功能角度进一步了解云架构。

1. 基础设施即服务

基础设施即服务（IaaS）交付给用户的是基本的基础设施资源。用户无须购买、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、网络资源和安全防护，如入侵检测、防火墙等。这些资源能够根据用户的需求动态分配。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，由用户根据需要进行使用，能够灵活适应用户动态需求。

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等。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。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，真正实现了云计算所允诺的“按使用付费”模式，从而能够节省用户的成本。

2. 平台即服务

平台即服务（PaaS）交付给用户的是丰富的“云中间件”资源，这些资源包括应用运行时环境、共享服务及自动化管理服务。平台即服务的应用运行时环境提供了最基本的应用运行的能力；共享服务，比如共享的数据库服务及消息中间件服务，能够为应用提供比传统软件更高的灵活性、可用性和安全性；自动化管理则为应用提供了快速部署和配置，以及服务等级协定（SLA）等管理方面的功能。

通常，平台即服务（PaaS）的用户可以大致分为三种：一种是平台本身的管理员，通过自动化管理服务了解平台的状态，从而保证平台的健康和资源的合理利用，以及进行相关的系统优化工作等；另一种是应用管理员，以应用为中心进行应用的监控和配置；还有一种则是应用开发人员，负责基于平台层的规范和共享服务进行应用开发。利用平台即服务开发应用和传统的开发模式相比有很大的优势：

- ▶ 由于平台即服务提供的高级编程接口简单易用，软件开发人员可以在较短时间内完成开发工作，缩短应用开发周期；
- ▶ 由于应用的开发和运行都是基于同样的平台，兼容性问题较少；
- ▶ 开发者无须考虑应用的可伸缩性、服务容量等问题，因为平台即服务提供了相应的保障；
- ▶ 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，支持开发人员使用Java和Python编写Web应用，并为应用所有者提供了应用的管理控制台。Google App Engine对Web应用无状态的计算和有状态的存储进行有效分离，并对Web应用所使用的资源进行动态分配，从而使该平台上托管的应用具有很好的可伸缩性和高可用性。

3. 软件即服务

软件即服务（SaaS）交付给用户的是定制化的软件应用，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，用户通过网络访问使用。软件即服务主要有以下三个特征：

- ▶ 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；
- ▶ 软件以服务的方式通过网络交付给用户，用户只需要打开浏览器或者某种客户端工具就可以使用服务；
- ▶ 采用了多租户技术，虽然服务面向多个用户，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在技术上都是一个巨大变革。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使用情况进行付费，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，而用户对于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。

7.2 云架构的特性

7.2.1 大规模

近些年来，信息技术和信息产业的发展见证了IT系统和应用的“大规模（Large-scale）”特征。通过互联网的传递，一个应用如搜索、社交网络、微博、科学计算等每天可以被遍及全球成百万、千万或者上亿的用户使用。为了支撑这样的应用，需要成百上千的服务器、TB或者PB级的存储空间及Gbps或Tbps级的传输带宽。例如，截止到2010年11月，Facebook有超过六亿的用户，总共上载了数千亿张照片，是世界上最大的照片共享服务。截止到2010年7月，微软的Hotmail有超过3亿的用户，它的搜索每个月要响应30亿次查询请求，它的Messenger也有上亿的账户，为了支持这样的业务，微软的数据中心有数十万

的服务器，而其中一半以上的服务器用于搜索业务。这样的应用和实现系统无疑是“大规模”这个词的有力注解。

“大规模”首先体现在IT物理设备的数量及IT应用所依赖的软件系统上。随着应用用户数量的增加，支持该应用的硬软件设备和软件系统也应随之增加。硬件的增加体现在更多的内存、更快的CPU处理能力、更大的带宽和传输速度、更多的存储空间和更快速的存储访问等。软件系统的增加体现在更多的计算和存储节点、有效的负载平衡、快速的故障检测和恢复、更多的层次结构包括缓存系统、业务与数据分离、边缘服务器等，大型分布式的文件系统和大型分布式数据库让数据更接近用户或者处理节点等。由于硬软件实体数量和层次结构的增加，大规模导致了IT系统的高度复杂性。

“大规模”也体现在IT系统在分布上的广域性。为了提高IT应用响应用户的及时性，也为了增强IT系统的可用性和可靠性，大型的IT应用通常采用分布式的系统结构，提供服务的服务器和存储空间分布在多个大型的、位于不同地域甚至不同洲际的数据中心中，这些数据中心通过宽带网络连接在一起。为了进一步让数据接近用户，大型IT应用还通过内容分发网络（CDN）将某些通用的数据提前发布并存储到全球各点成千上万的边缘服务器上。在这样的IT应用中，保持负载平衡、数据一致性和备份和及时的数据发布是系统设计和管理中需要重点关注的。

“大规模”导致了IT系统设计和管理的系列挑战。由于规模的增加，IT系统的结构变得越来越复杂，系统的运行和管理的难度也大为增加。需要特别注意的是，一些在小规模系统中微小的问题在大规模的系统就会变得显著起来。假设一个PC服务器的平均故障率是一年发生一次，在只使用一台服务器的情况下，用户一年仅需处理一次服务器故障，但如在有十万台服务器的数据中心中，每天则要处理两三百次的服务器故障。因此，涉及这样规模的IT系统，其容错能力和故障恢复能力及对于故障的管理能力都要得到特别关注。大规模系统设计中的一个基本前提假设是每一个构成系统的部件不是完美的、并非充分可靠的，需要通过系统结构和机制的设计来保证系统的可靠性和可用性。

在大规模的IT系统中，能量消耗已经成为关注的重点。有统计显示全球数据中心所消耗的电能超过了全球总电能的2%，在数据中心每年庞大的电能消耗中，实际只有不到一半的电能用在服务器等IT设备中。更严峻的是，由于服务器CPU平均利用率不到10%，这不到一半的能源又有大量的部分实际只用来维持服务器的空运转，没有产生任何实际效益。

因此，有效提高数据中心服务器等设备的利用率，将直接有效提升能量消耗的效率。

云计算在将IT资源和应用以服务的形式提供给用户使用，为了适应IT系统和应用大规模的特征，它需要有效应对“大规模”需求所带来的挑战。本书所介绍的云计算系统功能、架构和相关技术均以适应“大规模”特征为其出发点或者重要特征。

7.2.2 高可用

可用性（Availability）是指软件系统在一段给定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避免，云计算高可用性的本质是通过技术创新，保证即使软、硬件出现问题后服务仍然可用，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失效对业务系统的影响，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等服务也发生过无法访问的现象。这些问题的出现使得人们对现阶段公有云计算产品和服务的高可用性产生了疑问。

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，并通过提前准备副本、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

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，从而支持变化的服务请求。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所需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按需伸缩，即应用所占用的资源可以随着负载的增加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 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、存储和网络性能。但是，云环境所承担的计算、存储和网络方面的负载也远远大于传统的环境。不同的云环境所采用的技术可能不尽相同，本节将着重分析当前云环境中流行的技术的性能问题，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x86架构下主流的虚拟化平台中，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I/O性能开销问题则较为严重。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I/O密集型的应用，例如数据库，就会遇到性能问题。

作为云计算大规模数据处理的事实标准框架，MapReduce也存在性能问题。由于Google公司设计的MapReduce主要针对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，尤其是在传统的数据管理及数据处理的场景中，MapReduce缺乏数据库提供的很多核心功能，如索引、事务、视图等，也因此无法与很多现有的工具和系统进行衔接。

7.3 云架构的准则

7.3.1 信息安全与保密

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，二是云计算对现有的软件系统安全防护模式的挑战。

从云计算自身环境特有的安全问题来说，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，就是传统用户很难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。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”，而这是大多数安全厂商亟待解决的问题。

其次，在云环境中，部分用户的服务系统更新和升级大多数是由远程执行的，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；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。

目前应对云的安全问题，有一个传统的技术可以派上用场，就是虚拟专用网（Virtual Private Network，简称VPN）。VPN在云计算出现之前已经普遍地应用于企业网络应用中，一个典型的场景是，企业对自己的网络设置了防火墙和安全策略，在默认的情况下，如果用户处于企业外部的网络，由于他不在企业内部的网段之内，就无法访问很多企业内部受保护的服务。但在很多情况下，用户需要在家庭、宾馆、户外等场所接入企业网络以实现移动办公，这时就可以使用VPN。VPN会给这种移动用户提供一个虚拟的企业内部的网络地址，用户通过身份认证、授权等方式，利用这个虚拟地址接入企业内部网络，进行办公。VPN的这种使用场景在云计算的环境内可以得到很好的应用，云服务提供商及云用户可以设置灵活的访问控制策略，使得用户还是像在传统的局域网内一样地使用云，由此在一定程度上克服云计算的安全问题。

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。

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，甚至被加密存储。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。进一步说，用户还可以将数据以分块切割，然后以冗余备份的方式分别存储在多个不同的云存储服务上。

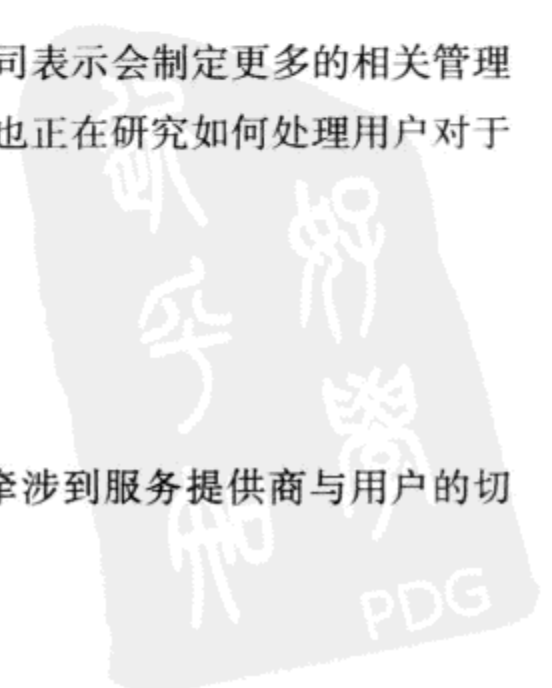
即便如此，上述方法只是增加了非法用户访问信息的难度，而没有从根本上解决问题。非法用户可以通过暴力破解所有的存储服务器来收集信息，他甚至会破解云存储系统的数据分发逻辑，从而精准地找到每一个块。同时，多个文件可能共同存储在一个大块里，这增加了数据泄露的风险。解决这些问题的根本做法是从逻辑上，甚至从物理上将多个用户的数据隔离。

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。

7.3.2 许可证与计费

许可证管理与计费涉及IT基础设施的最终支付环节，牵涉到服务提供商与用户的切



身利益。用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；各个提供商获得用户支付的费用。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数量、资源的规模、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被设想得非常美好，但是为实现这些设想还有很多工作要做。其中，最迫切的一个问题就是，大量的软件提供商目前还没有制定出其产品云计算环境下的计费模式。

目前比较成熟的云环境计费模型是Amazon公司提供的Amazon EC2和Amazon 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。

7.3.3 集成与标准化

越来越多的企业在构建自己的云计算平台，也有越来越多的企业和个人在使用云计算平台。从云计算平台提供者的角度来说，大部分云计算的平台是只能提供一类或者几类功能的，比如提供开发测试功能、海量计算功能，或者分析优化功能等。虽然Google、Amazon等公司试图提供一个大而全的云计算平台，但是他们的云计算平台提供的功能更多地是满足规模用户的需求，而不能完全满足用户的个性化需求。

从云计算用户的角度来说，由于不存在一个能完全满足用户所有要求的云计算平台，当用户使用云计算时，很有可能要同时使用多个云计算平台，而为了维护业务的连续性和一致性，用户需要维护多个云之间的数据同步、应用版本同步，或者使得应用在多个云之间能够互操作。应对这种需求，最理想的情况是多个云的操作方式、操作接口都是统一的，或者说这些云是同构的，那么用户就可以通过统一的操作方式来访问。但在大部分情况下，多个厂商提供的云是异构的，在这种情况下，就需要通过一种方法来将多个云抽象成一个，提供统一的操作方式和操作接口，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。

云计算为客户创造价值并不仅仅局限于公有云服务，比如通常要求客户将自己的核心业务迁移到公有云中是不可行的，一方面泄露商业机密的风险变大了，其中成本和迁移的不确定性也会在一定程度上增加业务风险，因此很多有实力的公司一方面将非关键业务迁移到公有云中降低运营成本；另外一方面在公司内部打造私有云来提高关键业务的服务质量和管理的自动化程度。很多时候公有云和私有云中的业务需要交换数据，这就需要一种集成服务将这些业务集成起来，如图7.2所示。

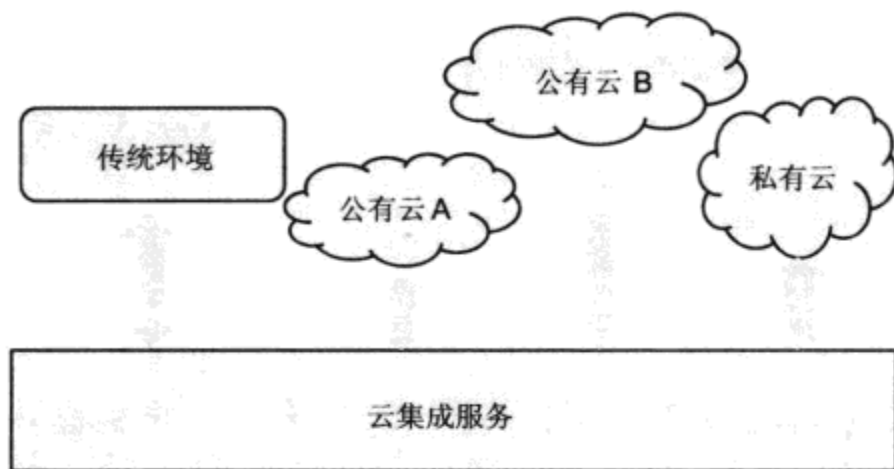


图 7.2 云集成平台

云集成服务需要VPN等技术将企业内外的网络链接起来。更重要的是，云集成服务需要访问云平台的API对不同云中的应用进行管理和数据访问。不难想象，如果没有云计算的标准化，那么云集成服务将会变得极为复杂，甚至无法通用而只能根据具体解决方案开发专用的云集成服务。这无疑会大大增加客户使用云计算的成本。前不久IBM收购了一家云计算集成公司Cast Iron，该公司能够帮助客户自动集成在Amazon和Google等云上运行的应用。

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，才会在这类厂商的带领下制定出相关的技术规范 and 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。

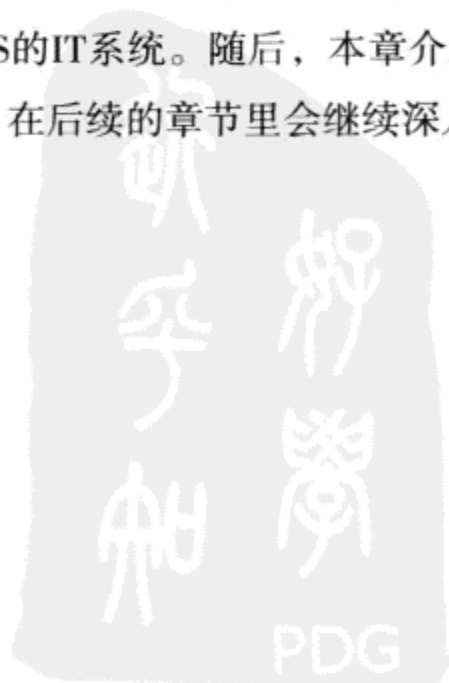
在云的基础设施领域，虚拟化的主要厂商之一VMware在发起一个叫做vCloud的接口规范，这个规范希望通过对基于虚拟化的数据中心（企业里的私有云）与其他“云”（其他企业的私有云、公共云）服务的接口进行标准化规范，来抢占云计算标准化的先机。在2008年9月举办的VMworld大会上，VMware公司宣布了它的vCloud计划，介绍了多家支持这个计划的厂商，并希望虚拟化的平台使用者、开发者甚至整个业界都来加入这个计划。VMware公司引领vCloud标准的优势在于它是虚拟化领域的先进厂商，拥有大量的合作伙伴，以及一个包括了大量应用的虚拟器件仓库。从本书的讨论中我们已经了解，虚拟化虽然是云计算重要的组成部分，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。

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设施，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。

7.4 小结

云计算将IT资源、应用平台和应用等以服务的方式通过网络交付给用户。为了应对网络时代对应用在大规模、可用性、可伸缩性和高性能的要求，云计算在系统架构上需要精心设计。本章从横向和纵向两个维度介绍了云架构。根据功能的依赖关系，云架构在横向上可以分为基础设施即服务（IaaS）、平台即服务层（PaaS）和软件即服务（SaaS）；根据企业IT系统对云计算的采用层次，云计算在纵向上可以分为传统的IT系统、基于IaaS的IT系统、基于PaaS的IT系统和基于SaaS的IT系统。随后，本章介绍了云架构的特性和准则，这些特性和准则将作为本篇的引子，在后续的章节里会继续深入展开分析。



第 8 章

基础设施即服务

- 8.1 概述
- 8.2 服务模型与接口
- 8.3 计算即服务
- 8.4 存储即服务
- 8.5 网络即服务
- 8.6 小结



基础设施即服务交付给用户的是基本的基础设施资源。用户无须购买、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运行软件系统和应用。基础设施即服务层向用户提供了虚拟化的计算资源、存储资源、网络资源和信息安全保障，如入侵检测、防火墙等。基础设施层根据用户的需求动态分配这些资源。相对于平台即服务层和软件即服务层，基础设施即服务层所提供的服务比较偏底层，能够灵活适应用户的需求。本章将深入基础设施即服务层，剖析其所提供的各项服务及它们的实现。

8.1 概述

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。Amazon EC2能够向虚拟机提供动态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，实现了云计算所倡导的“按使用付费”的模式。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。

简而言之，基础设施即服务提供的是虚拟化的计算资源、存储资源和网络资源。它们能够以服务的方式通过网络提供给用户来使用和管理。本章首先介绍实现基础设施即服务的服务模型和接口。服务模型是对资源的模型化描述，并通过模型接口暴露给用户使用，基础设施所包括的计算、存储和网络三种资源有其相应的抽象模型和接口。然后，本章将深入介绍计算即服务、存储即服务和网络即服务。对于每一种具体的服务形式，我们先会分析其特有的关键技术，再介绍一个参考实现，从而帮助读者更好地认识此类基础设施层服务。

8.2 服务模型与接口

为了向用户提供计算、存储和网络等服务，基础设施层需要基于物理资源灵活而动态地提供用户所需的能力，因而需要具备相应的基本要素。对内来说，基础设施层采用资源



抽象的方式整合各类异构的物理资源；对外而言，基础设施层通过服务模型和简化、易于操作的模型接口来供用户调用。下面分别介绍这两方面的基本要素。

谈起云计算，弹性、无限和服务是三个关键词。在基础设施层，弹性意味着计算资源能够根据负载进行调整，当负载高的时候分配更多的资源从而保证服务的质量，当负载低的时候释放多余的资源从而节省成本。无限意味着用户需要多少资源云平台都能够提供，并且能满足用户不断增长的需求。服务也是一个重要的特性，既然作为服务提供，基础设施层不仅要保证服务的功能，也要保证可用性、可靠性、安全性等和服务质量，还要为应用提供一些监控信息，如CPU的平均使用率，内存的占用有情况等。

这三个属性与模型接口的定义是息息相关的。弹性意味着资源要能够自由的伸缩，实现这个特性的一个重要方式就是降低系统之间的耦合性，而常用的方法就是模块提供独立的服务接口，而通过服务接口进行整合。由于数据中心的规模限制，某一个数据中心的资源往往是有限的。为了提供近似无限的资源，需要将同一个服务提供商的多个数据中心整合在一起，甚至是将多个不同的服务提供商的资源整合在一起。这需要统一的接口来对服务进行规范。

基础设施层提供的基本服务有计算即服务、存储即服务、网络即服务三类，实际中三者是密不可分的，一项服务可能会是这三种服务的组合，如一个对海量数据进行处理集群，计算、存储以及网络资源缺一不可。服务接口是基础设施层提供给网络用户或者应用的服务获取方式，它是平台提供功能的描述及对用户的一种承诺。通过服务接口，用户可以方便、快捷地从任何有网络接入的地方获取这些服务，如申请数十台到数百台的虚拟机，以及分配相应的存储，配置网络等。

如图8.1所示，基础设施层的核心资源包括计算能力、存储及网络。除此之外，为了管理这些资源，云、虚拟数据中心、虚拟机镜像、安全域、分组及用户等也是必不可少的组成部分。这些对象的操作也是通过接口来完成的。从图8.1上我们可以看到，一个云平台可以由一个或者多个分布在不同区域的数据中心组成，如Amazon EC2在不同的国家分别建立独立的数据中心为该区域用户提供服务。这样做一方面提高了服务的可靠性，解除了单一数据中心资源限制的瓶颈，另一方面也提升了用户的体验。如上一节所述，虚拟数据中心是采用虚拟化技术构建的提供云服务的数据中心，是资源的集合。它通过虚拟化平台屏蔽了底层物理资源的差异，提供了统一的运行环境。

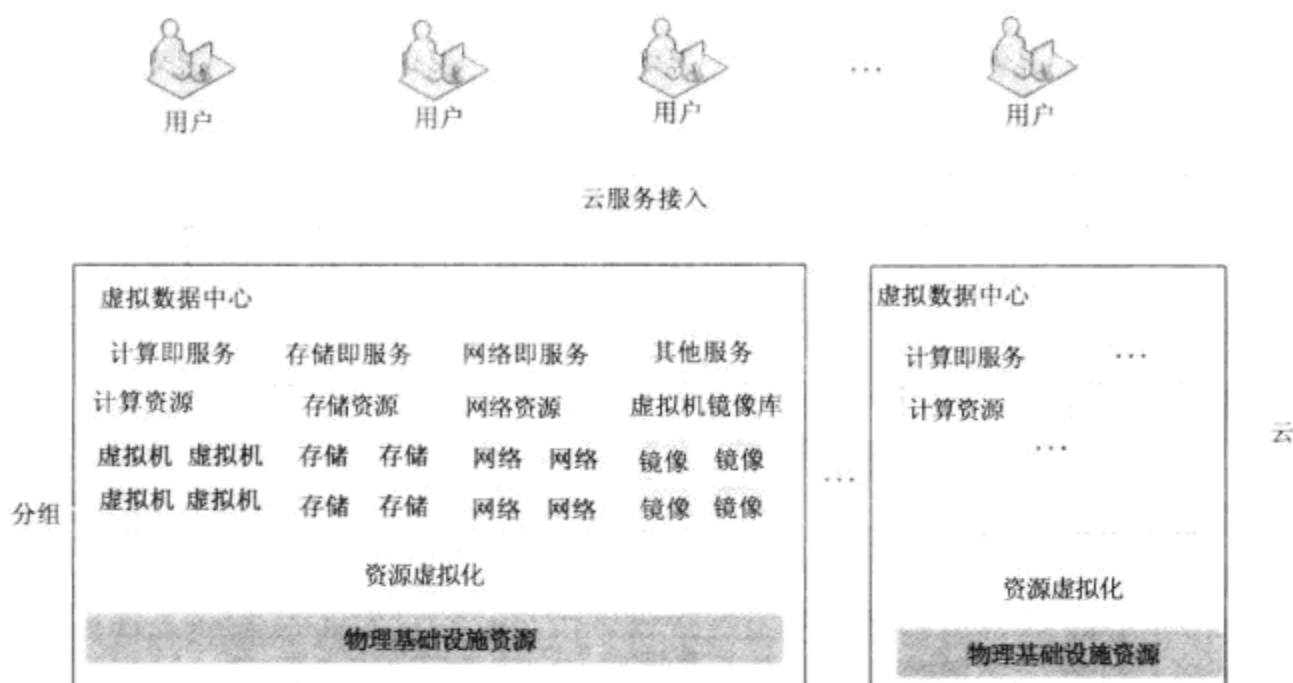


图8.1 基础设施层模型

为了减少虚拟机实例创建和初始化的时间，提高部署效率，我们通常根据虚拟机实例创建虚拟机模版，即虚拟机镜像。虚拟机镜像是二进制文件，是虚拟机实例的模版，包含创建虚拟机实例需要的所有文件和配置信息。使用虚拟机镜像可以创建任意数量的虚拟机实例。虚拟机实例是提供计算服务的资源，而对于日益增长的应用数据及日志等信息需要额外的空间进行存储，这就需要存储服务。存储服务为应用或者用户提供额外的存储资源。在基础设施层，存储资源常用的方式是为虚拟机分配的存储卷，用来存储应用的数据、日志或者虚拟机实例的备份等。网络服务为用户或者应用访问计算和存储资源提供了可能。它不仅负责网络信息的分配和回收，同时要负责网络层的安全，如VPN、VLAN和防火墙等安全策略。为了便于用户对一组资源进行统一的管理，基础设施层经常会引入分组的概念。以虚拟机实例为例，引入分组后，用户可以对一组虚拟机实例进行统一控制，如启动，停止或备份等。

在云中，服务是资源交付给最终用户的方式，所以了解服务显得尤为重要。这里的服务由三个要素组成：服务的模型、接口及服务质量。服务模型定义了什么是服务，如服务的组成、属性及特征。接口规范了服务支持的功能及相应的操作。服务质量是提供商和用户之间的协议，是服务供应商对用户提供服务的质量承诺。本章后续三节将有专门的部分分别从计算即服务、存储即服务及网络即服务方面介绍对应的服务模型和接口。

8.3 计算即服务

本节将分析基础设施层提供的基本服务之一，即基于服务器虚拟化技术的计算即服务。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，这使得计算资源的复用成为可能。

8.3.1 服务模型及接口

计算即服务是基础设施云提供的最核心的服务，即以虚拟机的形式将计算资源以服务的方式提供给用户。虚拟化技术的发展使得虚拟机资源的动态分配和调整成为可能，我们可以动态地根据业务的需求对服务器的资源进行调整。虚拟机带来的另外一个好处就是对同一物理机上的资源进行隔离，从而避免虚拟机之间相互影响。从云平台管理的维度来看，虚拟机因为能够带来管理的灵活性以及对底层异构资源的抽象而备受青睐，成为基础设施云交付的核心服务。

为了便于用户选择，每项计算服务都需要有一个详细的配置描述，包括软件的配置、网络和安全配置、硬件配置等。一般公有云会提供若干种配置供用户选择，如根据虚拟机实例的CPU、内存容量划分成的小型实例、中等配置实例、高配置实例甚至超高配置实例等。在私有云中，这种配置往往更加灵活，用户申请的实例的资源是可以任意配置的。从服务的生命周期来考虑，计算即服务要涵盖开发、部署、运行维护三个阶段，计算即服务的接口需要支持这三个阶段的管理。

1. 开发阶段

开发阶段也就是计算即服务的创建阶段，主要涉及对服务目录的更新，包括服务的列表、创建、更新及删除，对应虚拟机镜像文件和元数据的查询、上传、更新、下载及删除。一般情况下云服务商提供一组默认的虚拟镜像供用户选择，同时为了满足用户个性化需求，也支持自定义虚拟镜像的上传和更新。云服务接口可以支持不同的协议规范，本书以当前流行的REST协议为例进行介绍。从严格意义来讲，REST服务接口是由URI、请求参数及响应消息组成的。本节的主要目的是从功能上对接口进行介绍，并没有详细列出每个接口的请求参数和响应，如表8.1、表8.2所示。如果读者对这部分内容感兴趣，可以参考相关的实现文档。

表8.1 计算即服务开发接口

分类	方法	URI	说明
计算即服务开发接口	GET	/images	查看所有可用的虚拟机镜像
	GET	/image/VM01921699	查看ID为VM01921699的虚拟机镜像的详细信息，虚拟机的信息通常包括ID、名称、描述、状态、存储位置、类型及创建时间等。查询请求将返回这些信息
	POST	/images	创建新的虚拟机镜像
	PUT	/image/VM01921699	更新ID为VM01921699的虚拟机镜像的属性
	DELETE	/image/VM01921699	删除ID为VM01921699的虚拟机镜像。如果运行实例对这个虚拟机镜像的引用不为0，那么通常将虚拟机镜像标为不可用状态，直到所有引用的虚拟机实例被终止才进行删除

表8.2 虚拟机镜像列表查询接口示例:

请求消息示例:

GET /images HTTP 1.1

响应消息示例:

JSON响应消息示例

```
{
  "images" : [
    {
      "id" : 1,
      "name" : "SLES10 SP3",
      "updated" : "2010-12-10T12:00:00",
      "created" : "2010-09-10T13:08:00",
      "status" : "Tested"
    },
    {
      "id" : 743,
      "name" : "Redhat 5",
      "updated" : "2010-10-12T12:00:00",
      "created" : "2010-09-10T13:08:00",
      "status" : "Tested"
    }
  ]
}
```

2. 部署阶段

部署阶段的任务是对用户选择的服务目录中的服务模板进行实例化。对于计算即服务，目前一般是指利用虚拟机镜像创建虚拟机实例。实例化的过程一般包括下面几个阶段：（1）获取可用的虚拟机镜像（模版）列表；（2）选择一个虚拟机镜像及配置；（3）为虚拟机分配网络；（4）创建虚拟机实例并分配存储；（5）启动虚拟机。

对最终用户而言，只需要调用实例化的服务接口，选择包含指定虚拟机的服务，对实例数量和服务质量进行选择。在实例化之前还有个重要的阶段，即容量规划，来决策申请/分配多少资源给用户的业务。在云中，容量规划通常有两种方式：显式设定和隐式设定。显式设定是在创建实例前用户根据以往的经验或者性能测试软件计算得出资源的规模，然后设定一些资源调整规则，从而在负载有突发的变化时能够即时进行调整。隐式设定是用

户并不清楚初始的规模，只需要设定一个初始值，平台会根据用户要求的SLA进行资源分配和调整，而这一切由系统在后台完成。

计算即服务部署接口如表8.3所示。

表8.3 计算即服务部署接口

分类	方法	URI	说明
部署阶段	POST	/instances	创建新的虚拟机实例

3. 运维阶段

运维阶段主要涉及对虚拟机实例的运维管理，包括启动停止实例、对实例进行备份恢复、调整实例的资源、终止实例及监控等。云计算的一个重要特性就是自动伸缩，能够根据用户的需求动态调整资源以满足动态变化的负载的要求，从而达到资源的优化配置。资源伸缩涉及两个维度：垂直纵向伸缩和水平横向伸缩。水平横向伸缩在公有云中比较普遍，因为虚拟机的资源配置往往是固定的，如CPU类型数量、内存配置等。采用这种资源配置方式方便了管理且便于计费，在这种情况下资源的伸缩往往是通过增加或者减少实例的数量来实现的。垂直纵向伸缩是通过调整单个虚拟机的实例配置，即根据实例的负载动态调整CPU数量、内存等资源分配，这在私有云中经常被采用。为了支持伸缩，需要基础设施层提供查询接口，获取实例的状态，根据用户定制的或者平台提供的规则进行资源调整。此时，基础设施层需要提供调整实例资源配置的接口，来支持对资源进行的调整。当监控显示实例空闲时，基础设施层应该支持用户通过接口终止实例从而释放资源。表8.4给出了运维阶段的主要接口。

表8.4 计算即服务运维接口

分类	方法	URI	说明
运维阶段	GET	/instances	查询用户所有的虚拟机实例
	GET	/instances/redhat09900	查询id为redhat09900的虚拟机实例的详细信息
	PUT	/instances/redhat09900	更新id为redhat09900的虚拟机实例的配置信息。如调整CPU/内存、更改实例名称等
	POST	/instances/redhat09900	备份虚拟机实例redhat09900
	DELETE	/instances/redhat09900	删除虚拟机实例redhat09900

8.3.2 关键技术

8.3.2.1 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。

基于流传输的虚拟机部署方法通过COW（Copy on Write）的操作方式，使得虚拟机部署可以边传输边启动，可以有效地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8.2所示。

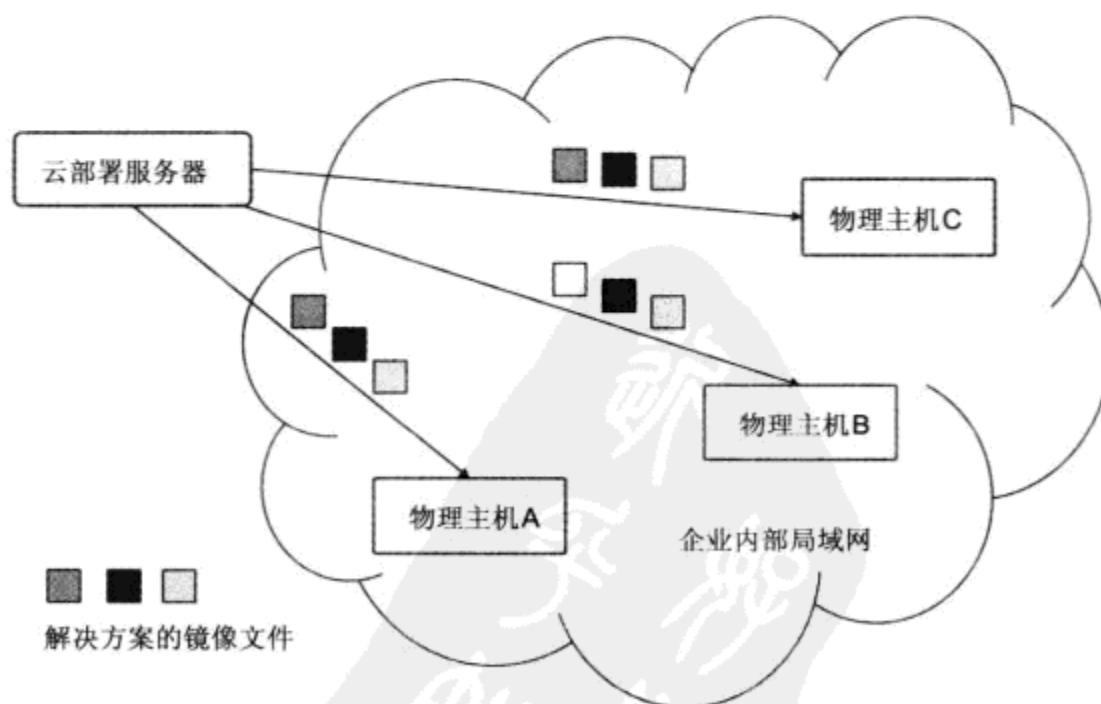


图8.2 并行部署系统架构

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使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，从而提高了部署速度。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，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8.3所示，协同部署系统的架构包括了部署服务器节点（图8.3中的云部署服务器）和被部署节点（图8.3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。

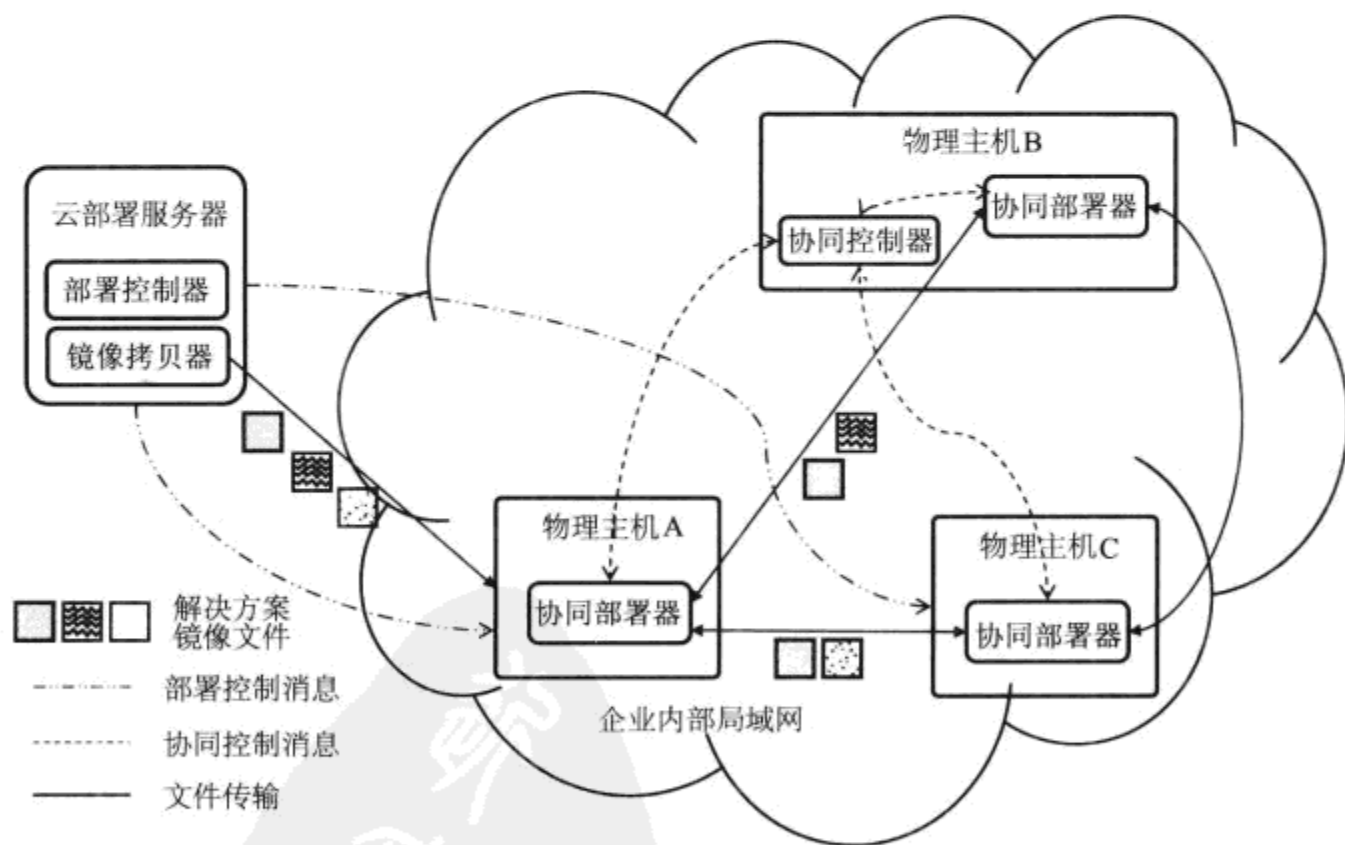


图8.3 协同部署系统架构

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。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包括两种情形：（1）在构建云平台环境并安装了服务器硬件后，需要在这些硬件上安装云的软件环境，这就涉及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；（2）在扩展云平台环境时（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，这种流程也被称为工作流（workflow）。在有些情况下，还需要对应的部署脚本。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，按照自动化流程执行任务或者执行部署脚本，驱动整个解决方案的安装配置过程。

8.3.2.2 资源调度

资源调度是指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（例如一个虚拟解决方案），每个计算任务在操作系统中对应一个或者多个进程。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分配给它的资源使用量，或者将计算任务转移到其他机器上。图8.4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。

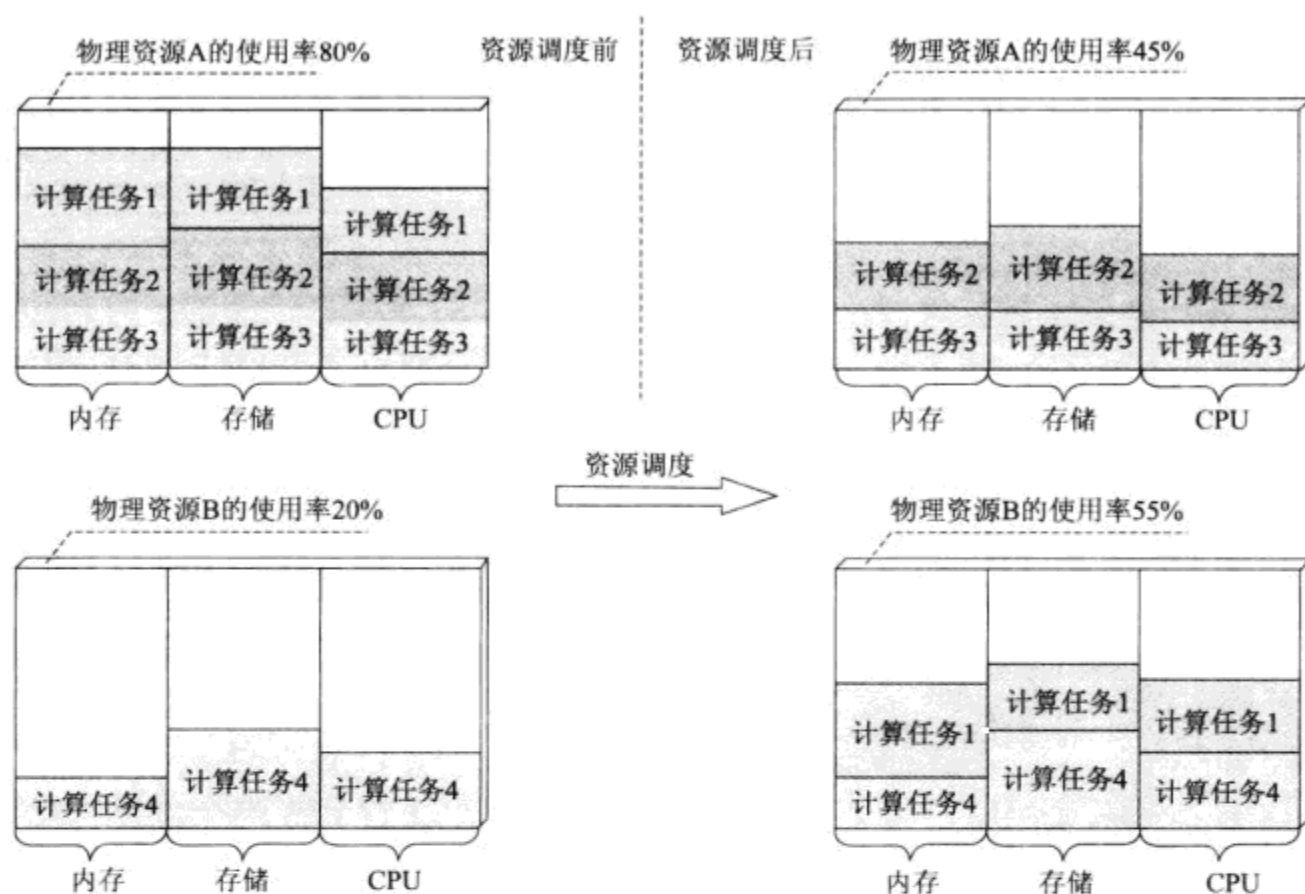


图8.4 资源调度

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。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（即虚拟机）进程从一台物理机迁移到另一台物理机，从而实现计算任务在不同物理机之间的迁移。

云计算的海量规模为资源调度带来了新的挑战。下面分别进行讲述。

首先，由于云计算环境中虚拟机的数量可能会很多，动态迁移在大规模环境下会遇到一些问题。第一，动态迁移要求迁移虚拟机的双方物理机共享使用一个存储服务，当虚拟机数量非常多时，存储服务可能会成为性能瓶颈，甚至无法提供服务。在这种场景下，需要动态迁移能够支持迁移双方分别使用自己的存储服务的场景。目前比较成熟的技术是VMware的Storage vMotion技术，它可以支持动态迁移时实现虚拟机镜像文件在不同存储服务之间的迁移。第二，当虚拟机迁移时，其网络配置是不变的，而在云计算环境中，网络配置需要灵活地进行调整，目前VMware已经意识到了这个问题，并在最新的vMotion中提供了对网络配置修改的支持。第三，目前动态迁移限制迁移的双方物理机处于同一个广播域内。在云计算环境中，虚拟机的数量非常大时，可能导致广播域无法给所有的虚拟机分配地址。针对这个问题，VMware推出了vNetwork Distributed Switch技

术，将多个广播域整合成一个虚拟的广播域，并维护所有虚拟机的地址。但这样的场景下还会有新的问题，比如广播风暴、安全问题等，目前业界还在努力解决这些问题。

其次，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。在这方面，主要依赖于云计算平台层的技术提供者能够提供详尽的资源使用情况数据。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，调度算法必须在精确性和速度之间寻找一个权衡点，或者提供给用户多种选择，是偏重精确性还是速度。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因素的限制而变得非常缓慢。

最后，从调度的粒度来看，虚拟机内部应用的调度才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性能目标进行比较，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。为了能够根据虚拟机内部应用的需求进行资源调度，需要有一套对于虚拟机内部应用的形式化记录方式，第6章中提到的“OVF标准格式”可以在一定程度上满足这种需求；另外，需要一套形式化的方法能够将应用的服务级别协定映射为一组资源调度的需求或者规则，这样，资源调度程序才能实现针对虚拟机内部应用需求的资源调度。

8.3.3 参考实现

8.3.3.1 系统设计

本节所展现的是一个计算即服务的实现，因为这是基础设施云的最核心功能，所以也可以称为一个简化的基础设施云。该基础设施云可被不同的企业使用，为了保证每个企业的安全性，企业网络都通过VPN访问基础设施云，所拥有的虚拟机的网络都会被配置为该VPN网内的网络地址。每个物理服务器上面都运行了若干个虚拟机，并且有一个系统管理器进行本物理机上虚拟机的自动化管理。每个企业内部还会有一个企业集群管理器，该管理器

负责在企业层次的资源管理，比如企业内部的资源划分和优化等。在基础设施云全局层次，还有一个云管理器进行云的整体监控和管理。该基础设施云的物理拓扑如图8.5所示。

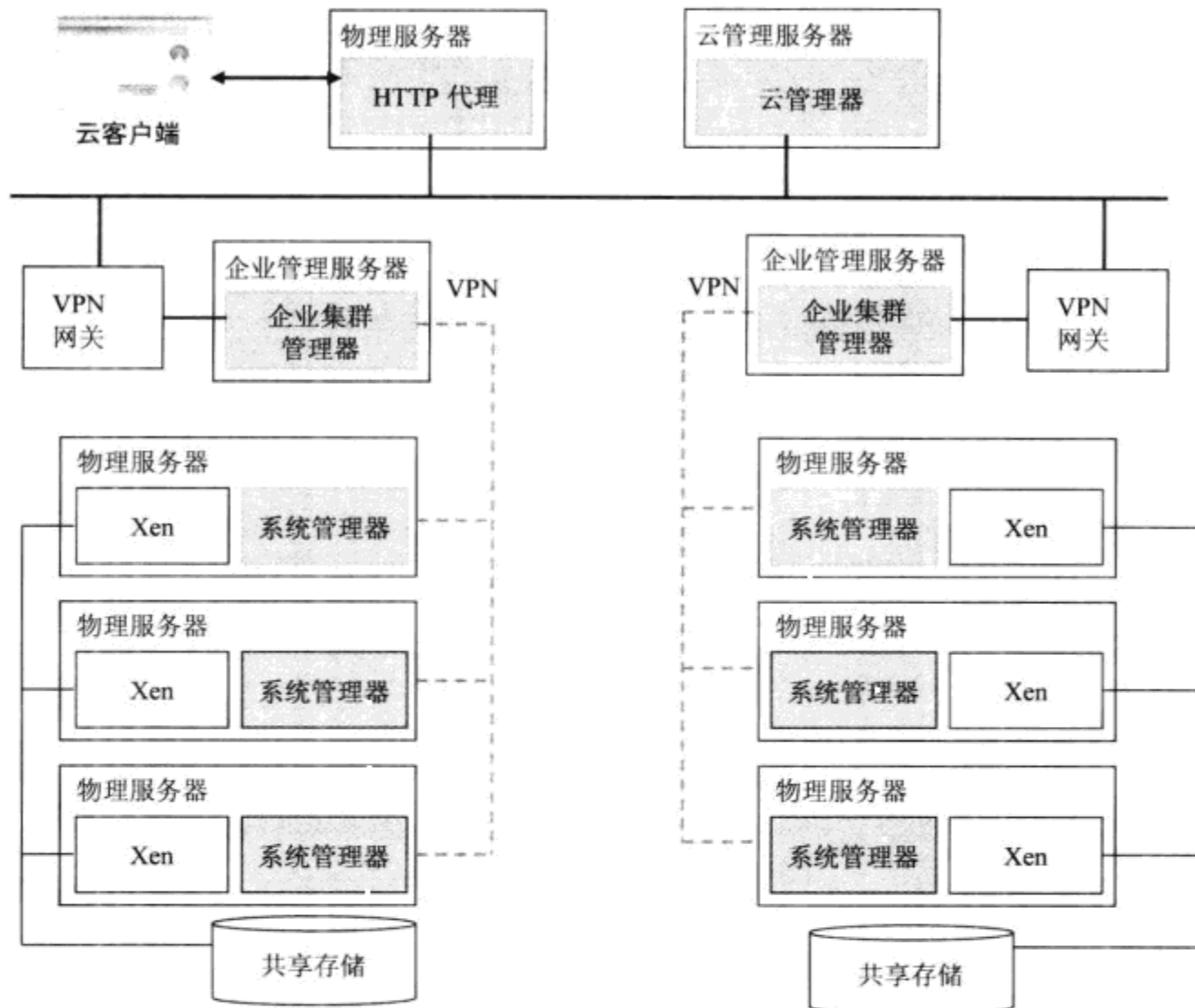


图8.5 基础设施层的物理拓扑

图8.5中每一台物理服务器都安装了Xen Hypervisor和一个系统管理器。系统管理器的主要功能是以独立于Hypervisor的方式提供各种必要的虚拟机管理功能，主要包括：（1）接收来自企业集群管理器的指令进行虚拟机的部署和启动/停止等操作；（2）监控本地虚拟机的可用性，在虚拟机出错不可用时尝试重新启动一个相同的虚拟机实例以替代失败的虚拟机；（3）实时监控各个虚拟机资源使用状况，根据监控策略将本地虚拟机的资源使用状况发送给企业集群管理器；（4）根据指定的SLA策略和虚拟机实时的资源使用状况自动执行虚拟机的伸缩操作；（5）接收来自企业集群管理器的虚拟机迁移指令，执行虚拟机的自动化迁移。

每个企业内部的物理服务器都共享同一个网络存储。每个物理服务器的本地存储只用来缓存每个虚拟机的运行实例，而虚拟机的镜像文件和实例状态都会被持久化到共享存储上来，这样能够更容易地支持高可用性和虚拟机的迁移操作。

每个企业内部的企业集群管理器的主要职责是企业级别的基础设施管理，包括：

- (1) 企业的安全管理，创建企业基础设施管理员帐户并设置管理权限；
- (2) 在共享存储上管理企业内部的虚拟机镜像；
- (3) 在企业内部进行资源规划和管理，比如决策如何通过服务器整合提高企业资源使用率；
- (4) 虚拟解决方案的管理服务；
- (5) 为云管理器提供企业级别的资源使用信息。

该基础设施云有一个云管理器，其主要职责是为基础设施云的提供者提供云的整体监控、管理和计费等功能，主要包括：

- (1) 集中统计每个企业的资源使用情况；
- (2) 根据每个企业的资源使用情况进行计费；
- (3) 提供整个基础设施云的SLA监控，包括实时的网络流量、各项管理操作的性能及云的可靠性等。

下面将着重介绍企业集成管理器如何进行虚拟解决方案的管理。虚拟解决方案由一组预先安装和配置好的虚拟机构成，这组虚拟机运行起来可以提供特定的应用或者服务。构成虚拟解决方案的每个虚拟机内部都置入了代理（Agent），用来监控和操作虚拟机内部的软件和服务。如图8.6所示，企业集成管理器都有一组虚拟解决方案管理服务，包括负载管理、资源部署、安全管理、数据管理、资源监控。虚拟解决方案管理服务通过与系统管理器和虚拟解决方案中的Agent进行交互，可以实现对虚拟解决方案的黑盒管理和白盒管理。企业管理员通过虚拟解决方案管理服务可以更好地对企业各个虚拟解决方案进行SLA管理。

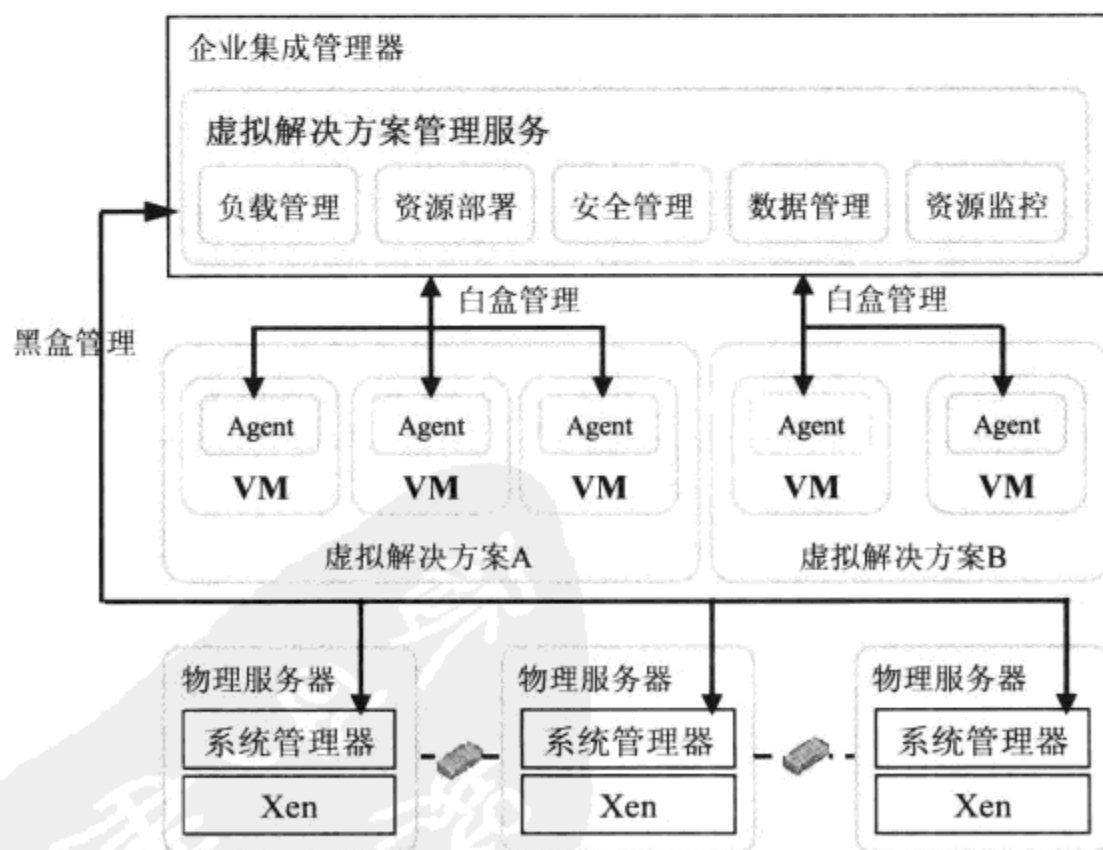


图8.6 虚拟解决方案管理

下面将通过一个使用基础设施即服务的典型流程来系统地介绍该企业集成管理器所提供的各种虚拟解决方案管理服务。

典型的基础设施层服务流程分为规划、部署和运行三个阶段，如图8.7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资源提供的功能，从而支持用户通过系统管理服务进行应用系统部署和卸载。部署阶段过后就进入了运行阶段。在这个阶段，用户的应用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计费服务。下面以这三个阶段为线索，向读者介绍基础设施层的基本功能和为用户提供的服务。



图8.7 基础设施层服务应用流程

8.3.3.2 系统实现

1. 规划阶段

在基础设施层的物理环境已经准备就绪后，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，企业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操作资源。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。用户通过这个过程获取现有资源列表，选择其所需要的虚拟机类型，实施部署和运行管理等操作。

该示例中企业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，虽然大小在若干GB量级，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。企业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。

2. 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启动起来。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描述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后，虚拟机内部的Agent将接收到企业集成管理器的激活指令。Agent根据OVF描述文件启动和配置这个实例，使虚拟机内的软件和服务进入运行状态。

基础设施层在企业集成管理器、系统管理器和OVF描述文件的帮助下可以实现高度自动化的解决方案部署，并且简化应用系统的激活过程。基础设施层提供给用户可视化的OVF描述文件编辑界面，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单击“开始”按钮一样简单。

3. 运行阶段

如图8.6所示，每个虚拟机内部安装的代理可以用于对虚拟解决方案进行更有效的运行

时SLA管理，代理接收企业集成管理器的指令，实现对虚拟机内部软件的管理。企业集成管理器以两种方式对每个虚拟机进行管理：

- ▶ 黑盒式管理，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。这种方式是通过企业集成管理器与系统管理器的直接交互完成的。
- ▶ 白盒式管理，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企业集成管理器与虚拟机内部的代理之间的交互来完成的。

4. 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，如果不能对资源进行有效的监控，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频率。对于网络，则需要对网络实时的输入、输出流量、可获得带宽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对象及其度量。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包括相互关联的多个虚拟资源的某个具体的解决方案整体进行资源监控。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某个解决方案整体资源的使用情况及其对解决方案整体性能的影响，从而采取必要的操作对解决方案进行调整。

资源监控是通过企业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，企业集成管理器通过与系统管理器的通信，获得每个虚拟机运行时资源监控信息。通过对单个虚拟机资源监控信息的进一步分析整合，企业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，企业集成管理器需要管理的是虚拟机内部的软件栈。代理负责接收企业集成管理器的状态监控指令，根据该指令获取虚拟机内

部软件的运行状况监控信息，然后将这些监控信息发给企业集成管理器。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企业集成管理器提供的特性。

5. 负载管理

在基础设施层这样大规模的集群资源环境中，任何时刻参与节点的负载都是起伏不定的。图8.8左所示为负载管理之前某个时刻负载分布状况。

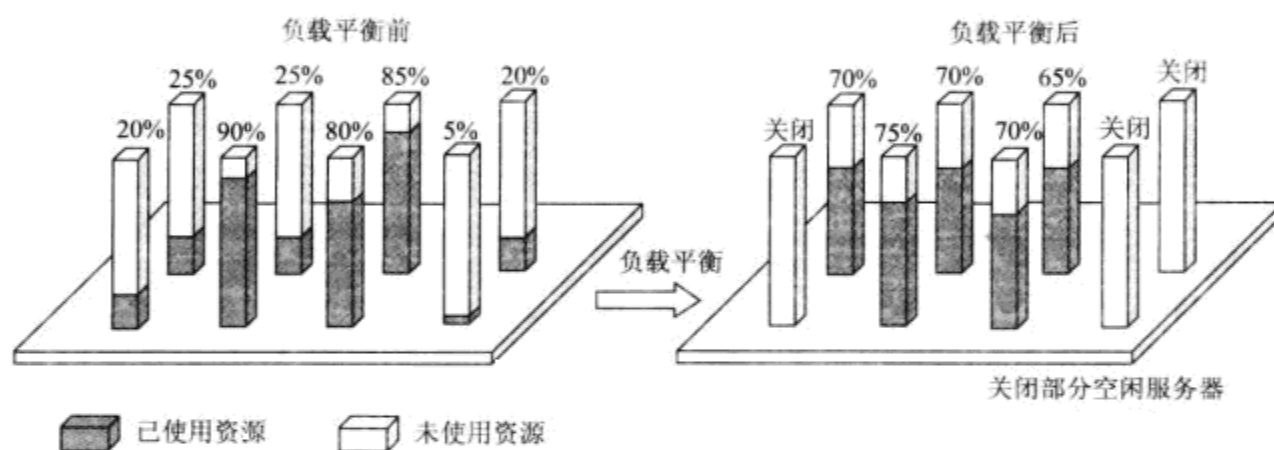


图8.8 负载平衡效果示例

一般来说，节点之间的负载允许存在一定的差异和起伏，它们的负载在一定程度上不均匀也不会导致严重的后果。可是，太多节点资源利用率过低或者节点之间负载差异过大就会造成一系列突出的问题。一方面，如果太多节点负载过低，会造成资源使用上的浪费，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部分负载到负载过低节点，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。图8.8右所示即为负载平衡后的负载分布状况。

负载管理是基于资源监控功能来实现的，并且同样依赖于企业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，企业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，可以通过调用系统管理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，企业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企业集成管理器。由此可见，代理在白盒管理中承担了企业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。

6. 安全管理

在云计算模式中，用户因为失去了对硬件资源和数据的控制而对云计算的安全性抱持怀疑的态度。Gartner发布的一份名为《云计算安全风险评估》的报告列出了云计算技术存在的七大风险。即特权用户的接入、可审查性、数据位置、数据隔离、数据恢复、调查支持和长期生存性。

Gartner报告中所提到的云计算环境中存在这些风险，并不是说在传统环境中就不存在，而是说有些风险在云计算中更为明显或更有挑战性。以长期生存性举例，传统方式中，一个客户买了某个软件提供商的产品时，并不是说将来这个软件提供商就一定不会倒闭或者被兼并，只不过由于软件产品市场相对成熟，通常可以比较容易地找到替代产品，从而相应的风险是可控的。

由于目前云计算还处于初级发展阶段，很多服务都没有标准化，因此市场上能够相互兼容相互替代的产品还不是很多。在这种情况下，云计算服务长期生存性的风险就相对更大了。对于用户而言，在选择云计算服务的时候，只要对可能的风险有充分的认识并制定好应对措施，风险是可以很好地控制的。比如数据恢复风险，客户需要咨询清楚云计算提供商是否能够应对数据损失风险及相应的恢复措施，也需要明白在发生灾难时自己的数据和服务会发生什么事情。客户也需要知道在风险发生时应采取何种数据恢复措施从而避免损失。

在基础设施层，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访问入侵。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和操作是安全的，并提供安全审查机制保证对云中数据的操作都是经过授权的并且是可被追踪的。

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。对于本节介绍的简化的基础设施层示例来说，企业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。企业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。

7. 计费管理

云计算作为一种信息技术与传统的方式相比，只有让用户以更低的成本获得更高的价值才能够得到用户的认可，从而蓬勃发展起来。因此，合理的计费模式非常重要。云计算倡导“按使用计费”的模式。

目前，面向公共提供服务的基础设施云主要的计费模式是依据某个时间段内应用所消耗的存储、网络、内存等资源向用户收费。实现这种计费模式有两种方式：（1）实时计费，真正用多少资源就收多少费；（2）为用户预设几种固定资源配置类型，每种类型都有单位时间内的计费标准，用户选择一种类型使用并根据使用时间长短进行付费。第一种方式对各种资源使用的监控准确度要求更高，因此很少被真正采用；而第二种是一个相对折中的办法，使用时间长短和所选择的资源配置类型作为计费依据，实施起来更加容易，因此被广泛采用。

实际上，计费管理不仅仅是基础设施层所需要的，而是云计算不同的服务模式都需要的。当然，在不同的层次和服务类型上，计费所选择的变量是不同的，而相同之处在于都需要依据对相应服务的使用时间进行费用计算。例如，在平台层计费是以对平台的使用为核心的。一种计费方式是基于应用程序对资源的使用状况进行计费，比如Google App Engine，计费方式与基础设施层类似；另一种计费方式是基于软件的许可证的。与传统的许可证计费方式略有不同的是，新的许可证计费模式会结合基础设施层资源的使用状况来制定计费标准。比如表8.5列出了IBM软件在Amazon EC2上的计费标准。表8.5表明，IBM软件在EC2上的计费标准是与EC2实例类型相关的。

表8.5 IBM InfoSphere DataStage/QualityStage在EC2上的定价

EC2实例类型	资源区域：美国东部 (N. Virginia)	资源区域：欧洲 (Ireland)和东南亚 (Singapore)
Standard Large	\$18.92 per hour	\$18.96 per hour
Standard Extra Large	\$36.91 per hour	\$36.99 per hour
Double Extra Large	\$37.43 per hour	\$37.57 per hour
Quadruple Extra Large	\$75.63 per hour	\$75.91 per hour
High-CPU Extra Large	\$73.91 per hour	\$73.99 per hour

来源：<http://aws.amazon.com/ibm/>。

资源监控和负载管理是为用户提供计费、运行监控服务的基础。通过对虚拟机的配置、使用时间及服务质量的综合考虑，基础设施层为用户提供精确的计费服务。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。

8.4 存储即服务

在日新月异的信息时代，人们正在产生和处理海量的数据。根据一项调查显示，2006年全世界的总数据量大约是140EB（1 EB=1 000 000 000 000 000 byte），预计到2011年将会超过1.8ZB（1ZB=1 000EB）。存储是数据的载体，面对呈爆炸式的数据增长，提供高可靠和高可用的存储服务是一个不小的挑战，仅靠孤立的单个或者多个服务器的本地磁盘存储显然已经无法满足这样的需求，人们更多地是采用网络存储来解决这个问题。

网络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、吞吐率和响应速度等各方面要求。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需要存储海量数据，比如搜索引擎和互联网视频网站，这些需求催生了一系列优秀的分布式网络存储技术。

从云存储即服务的角度来看，主要给用户提供了三种形式：（1）提供块级别的存储服务，比如Amazon EBS；（2）提供文件级别的存储服务，比如Amazon S3；（3）提供结构化数据的存储服务，比如Amazon SimpleDB。

在下面的小节中，我们将以块级存储为例讲解其访问接口，然后介绍支撑大规模网络存储所使用到的关键技术，包括存储区域网络（SAN）、网络接入存储（NAS）和分布式文件系统。其中，存储区域网络提供的是块设备存储服务，网络接入存储和分布式文件系统提供的是文件级别存储服务，至于结构化数据的存储服务，本书中将其归为云平台的关键技术，将在后面的章节中讲解。最后，在参考实现这一节，将介绍Google File System（GFS）。

8.4.1 服务模型及接口

存储即服务是为用户或者应用提供数据存储的服务。在基础设施层上，它通常是为用户虚拟机实例提供额外的存储，如块存储设备或者文件系统。Amazon EBS服务是商业化存储即服务的实例。通过采用存储服务，可以将应用产生的日志、数据进行持久化，采用灵活多样的方式访问云中的数据，而无须关注虚拟机实例的状态。

基础设施层提供存储服务的优势在于：（1）数据存储和计算的分离，降低了系统间的耦合性，提高了系统的可靠性，计算服务服务器的失效不会影响对数据的直接访问；（2）提高了系统的可扩展性，存储作为单独的服务提供，可以灵活地根据需要动态地进行资源调整，而这一切对计算模块是透明的；（3）按需付费，根据使用的资源大小进行计费；（4）降低成本，通过专业化的公司提供服务，通过规模化可以减少用户的成本。

为了构建存储即服务，平台构建者首先需要采用虚拟化技术将所有的存储资源组成一个大的资源池。通过构建资源池可以屏蔽底层异构存储的差异，建立统一的存储模型，从而对资源进行统一的管理。这样不仅可以简化资源的分配回收及更新，而且便于对资源池的扩展和管理维护。

在基础设施层，存储即服务提供的主要功能有：申请新的存储，将存储划分给虚拟机实例，调整虚拟机的存储，回收存储资源等。将存储作为服务提供时，存储对象一般包含以下属性：卷组的名称，对卷组的简单描述属性，存储卷的格式，存储卷的大小，存储卷的状态（如当前存储卷是否同虚拟机实例绑定及是否正被使用等），以及创建时间等。表8.6简单列举了将块设备作为存储即服务的主要接口。

表8.6 存储即服务接口

方法	URI	说明
GET	/volumes	查询用户所有的存储卷列表
GET	/volumes/vol09900	查询id为vol09900的存储卷的详细信息
PUT	/volumes/vol09921	更新id为vol09921的存储卷的信息，如描述、大小、格式等
POST	/volumes	创建新的存储卷
DELETE	/volumes/vol09921	删除存储卷vol09921

8.4.2 关键技术

8.4.2.1 存储区域网络

存储区域网络（SAN，Storage Attached Network）在业界已经有几十年的历史，它经典、成熟而又在不断地发展。简单来说，SAN是一种高速存储网络，它负责提供服务器和存储系统之间的串行SCSI数据传输。从SAN的系统构成来看，它主要分为存储端、网络和服务端三个部分，如图8.9所示。

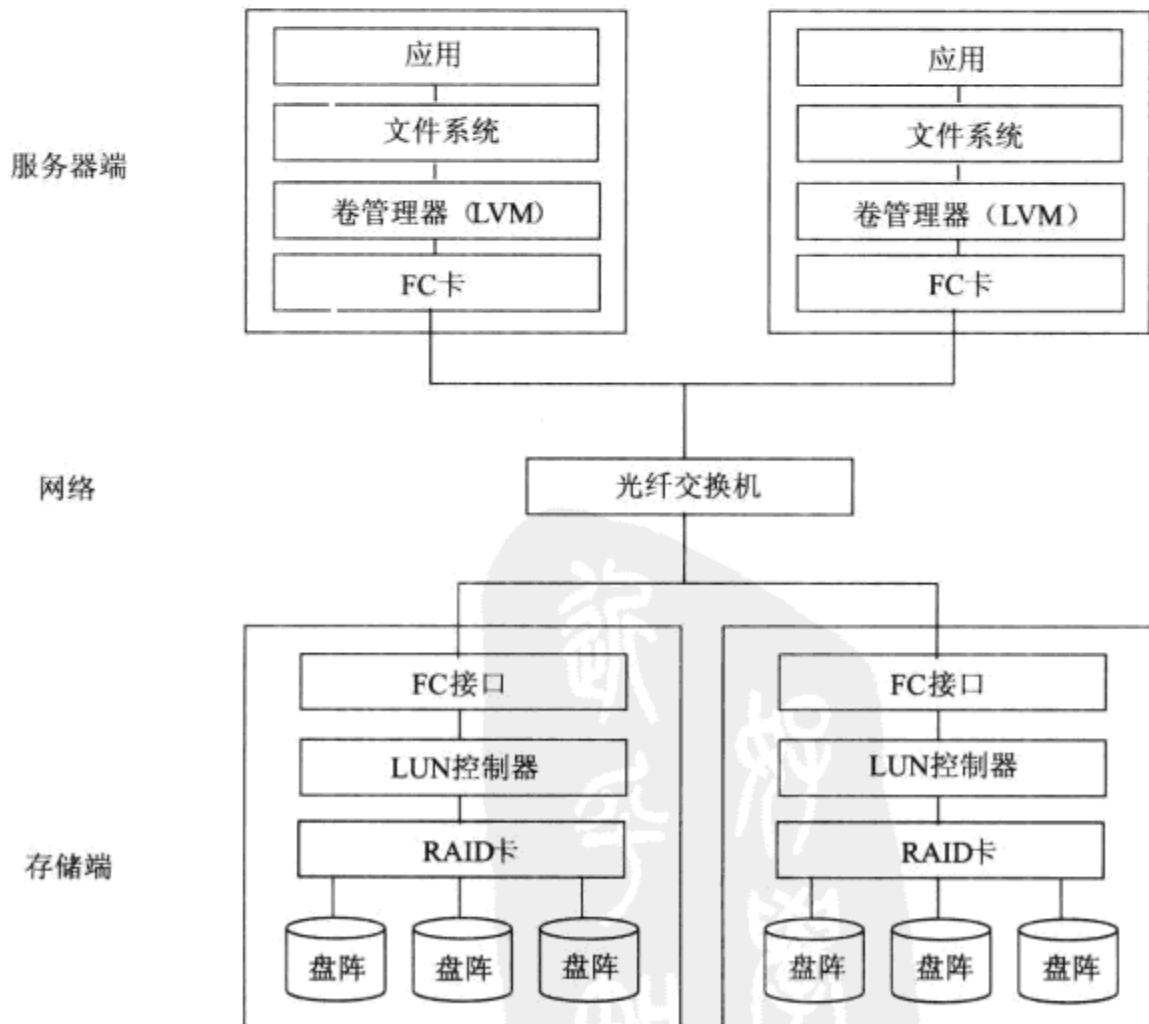


图8.9 SAN基本架构

存储端通常是由一组或多组磁盘构成的磁盘阵列。磁盘阵列的管理器通过RAID技术实现，包括对磁盘的管理及数据的冗余备份和校验，从而保证了数据的可靠性和安全性；管理器通过LUN分割和掩码技术实现对底层的磁盘虚拟化，形成由若干逻辑磁盘组成的存储池，从而可以共享给更多的服务器；最后，存储端提供了和服务器的通信接口，实现内部I/O协议和外部I/O协议的转化，比较常见的是采用光纤通道卡和以太网卡，从而实现了外部可访问性。

网络是SAN系统的核心，也是区别于传统内置式存储方式的重要特点。在计算机历史发展的初期，磁盘存储是服务器的内置一部分，磁盘通过SCSI总线 and 控制器与服务器系统的南桥芯片驳接，这种方式被称为DAS（Directly Attached Storage），现在普通个人电脑仍然采用这种方式。不过，SCSI总线最多只能直连15个磁盘，随着数据量的日益增长，DAS方式已经无法满足人们的存储需求，于是诞生了服务器和存储分离的想法。服务器和存储分离以后，通过专用网络的方式连接在一起，存储端可以不受服务器的限制，而服务器可以通过网络共享同一个存储设备，这也是SAN的雏形。在DAS系统中，磁盘数据和控制指令的传输都是基于并行SCSI协议来实现的，而并行SCSI协议无法实现远距离的数据传输，改造成存储网络必须将并行SCSI协议转化成串行方式。其中，高端的串行SCSI方式是采用光纤通道（Fiber Channel，FC）作为底层传输协议，存储端和服务器端通过FC卡连接到光纤交换机上，采用包交换方式传输SCSI协议，实现数据通信。不过，构建FC SAN的相关设备非常昂贵，这也限制了它在业界的普及。近年来，人们开始采用廉价的以太网和IP网络来承载SCSI协议，并且形成了iSCSI协议标准，SAN才开始慢慢走向平民化。

由于服务器和存储的分离，服务器需要通过光纤通道卡或者以太网卡来接入存储网络，从而实现数据存储和传输。在服务器操作系统中，存储端分配的逻辑磁盘以块设备形式暴露出来，通过卷管理器（LVM）挂载到文件系统上。

经过多年的发展，SAN已被广泛应用在企业级存储中，它可以支撑TB级乃至PB级的海量存储。在云计算时代，很多企业的私有云就采用了SAN作为存储即服务的支撑技术，它可以非常方便地实现虚拟化的存储资源池；在公有云领域，Amazon EBS就是类似于SAN这样的实现，它向Amazon EC2实例（虚拟服务器）提供块级别的存储。

8.4.2.2 网络接入存储

和SAN一样，网络接入存储（Network Attached Storage，NAS）也是一项传统的技术，

不过它和SAN最大的区别在于提供的存储服务不同，SAN是面向块级别的，NAS是面向文件级别的。NAS系统的组成可以分为三个部分：存储端、网络和服务端，如图8.10所示。

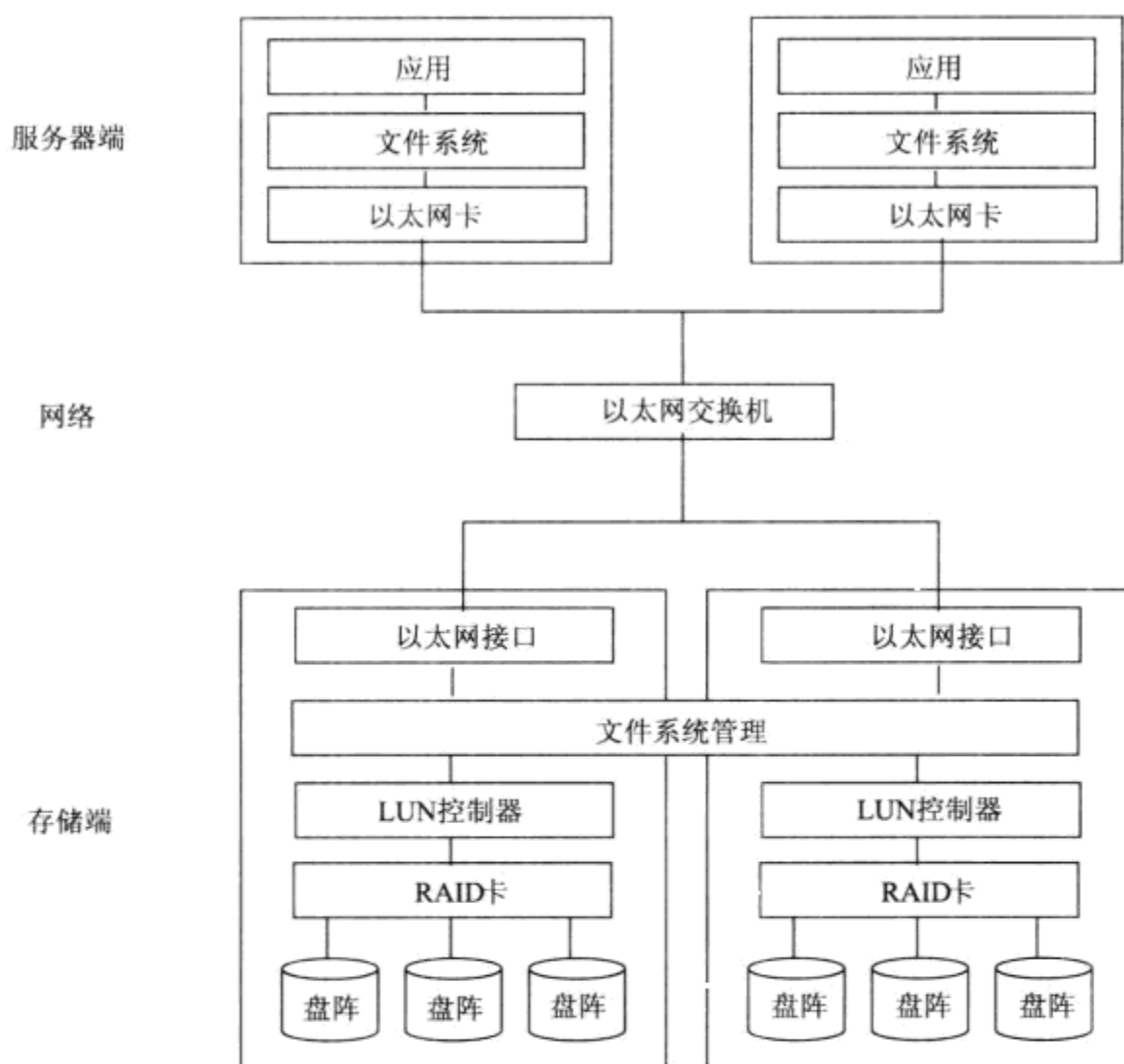


图8.10 NAS基本架构

NAS的存储端主要是由虚拟化的存储资源池、控制器和接口组成的。存储资源池的具体实现一般是采用磁盘阵列，也可以采用SAN。NAS存储端的控制器功能比较强大，它拥有独立的处理器单元、内存、I/O处理模块和文件系统管理功能，负责将磁盘阵列的虚拟磁盘挂载到自己的文件系统上。通过这个文件系统，NAS存储端可以很好地管理数据共享，避免读写不一致，这也是和SAN的重要区别之一。存储端的接口负责将内部的文件I/O和网络I/O连接起来，并且通过以太网卡将该文件系统暴露给外界。

NAS的网络协议栈以网络层为界，分为上下两层。下层的实现有以太网/IP、ATM或FDDI，通过网络交换机实现互联，上层采用NFS或CIFS协议实现文件级网络数据共享。在NAS的初期，带宽一直是该技术性能上的瓶颈。随着IP技术的发展及千兆以太网乃至万兆以太网的出现，NAS的性能得到极大改善。另外，NAS的网络采用廉价而技术成熟的以太

网/IP技术，并不需要专门的存储网络，可以很容易地被中小型企业接受。至于上层协议，NFS和CIFS都提供了基于网络的文件系统，它和存储端的控制器配合能够支持文件和数据的共享和并发读写保护。

由于NAS存储端有专门的处理控制器来维护和管理文件系统，服务器端变得非常简单，往往只需要普通的以太网卡及操作系统支持NFS或者CIFS协议，最终NAS的网络文件系统通过某个挂载点被挂载到服务器虚拟文件系统（VFS）上。

最后，要让NAS支持云存储服务，还需要赋予它可扩展的能力来支持大规模的数据访问和读写。具体而言，需要对NAS存储端的控制器及文件系统进行改进和优化，来支持底层磁盘阵列的动态扩展及维护统一的命名空间。

8.4.2.3 分布式文件系统

我们知道，文件系统是负责存储和管理文件内容和元数据的软件系统，本地文件系统是将文件内容和元数据都保存在本地服务器上，而分布式文件系统是通过网络将这些数据保存在多个远程服务器上。通过分布式文件系统提供的接口，用户可以透明地操作存储在网络上的文件。

相对于传统的文件系统，分布式文件系统具有很多优势。首先，由于不受本地存储空间的限制，它可以通过动态增删存储服务器来实现高伸缩性；其次，数据可以在多台服务器上进行冗余备份，提高了分区容忍性，避免了单点失效，提高了可靠性和可用性；再次，支持大规模数据访问，由于每个用户读写数据只需要在特定的几台存储服务器上进行，所以总的工作负载被分散到所有的存储服务器上，不会存在单点性能瓶颈；此外，分布式文件系统也能提供多用户/组的访问控制、存储配额和文件加密等特性。

根据分布式文件系统的架构，其大致可以分为三类：客户机/服务器架构、主从式集群架构和对等式集群架构。客户机/服务器架构最早出现，它通过网络实现了文件系统客户端与文件存储和管理的分离，其典型代表是Sun公司开发的NFS文件系统，也是目前使用最广泛的分布式文件系统之一。主从式集群架构是客户机/服务器架构的增强，它实现了客户端、元数据存储和管理与文件内容存储和管理三者的分离，主节点负责维护和管理文件系统的元数据和管理，比如命名空间和存储节点，若干个从节点负责维护和管理文件内容，客户端与主节点通信获取需要访问文件的相关信息，比如存储文件内容的从节点的地址，

然后客户端和相关从节点通信，访问文件具体内容，这类架构的典型代表是GFS，稍后我们还将对它进行重点剖析。对等式集群架构中，没有了主节点，每个节点都参与管理和存储数据，节点的底层采用了分布式哈希P2P存储系统，实现了数据存储持久化，节点还维护了相应的操作日志，通过查找所有节点的操作日志实现了读数据的一致性，在节点上写数据是通过追加方式记录在自己的操作日志上，这样就避免了使用全局锁，这类架构的代表是由麻省理工大学开发的Ivy分布式文件系统。

分布式文件系统的组件按照功能可划分为：接口（客户端）、文件内容存储及管理以及元数据存储及管理三个方面来介绍。

分布式文件系统提供的接口既可以是和服务器端本地文件系统兼容的，也可以是不兼容的。目前，主流的操作系统基本都采用了抽象的文件系统接口，分布式文件系统只要定义了这些抽象文件系统所要求的接口操作，就能够实现和本地文件系统的兼容，比如NFS就能够挂载到Linux的VFS文件系统上，用户就可以像访问本地文件系统一样来访问网络文件系统。当然，采用私有接口的也不少，通常是命令行、应用编程接口开发包或者Web服务，比如GFS和HDFS采用了应用编程接口和命令行，而Amazon S3采用了Web服务接口。

元数据存储和管理包括了文件系统命名空间、访问控制信息、文件与块之间的映射关系及节点联系地址。文件系统命名空间维护了文件系统的目录结构和文件与目录的位置关系，对文件的创建和删除请求都会影响命名空间，因此需要有相应同步的机制来处理多个请求，比如用读写锁。访问控制信息规定了哪个用户或组拥有对文件或目录所有权和读写权。在存储文件的时候，分布式文件系统通常会将文件切割成若干个块，然后将每一个块复制到多个存储节点上，这样既可以实现文件的冗余备份，也能够分散对文件不同部分的读写请求，因此负责元数据存储和管理的节点还需要给文件块编号索引和维护文件与块之间的映射关系，以及协调如何放置文件块到存储节点上。负责元数据存储和管理一般是主节点，它需要知道从节点（存储节点）的联系地址，从节点启动的时候会向主节点主动汇报，以后通过心跳信息保持更新。

文件内容存储及管理包括文件内容的存储和校验，向负责元数据存储及管理的节点汇报节点信息和存储信息，以及和客户端通信实现对文件内容的读写操作。通常负责文件内容存储及管理的节点会将文件块存储在自己本地的数据持久化层中，但也可能将其存储在网络上，同时它还需要负责对文件块生成校验值，用来检测文件块是否损坏，这在建立在普通PC架构上的分布式文件系统尤其重要，因为底层硬件随时可能发生故障。此外，负责

文件内容存储及管理的节点还需要定期向元数据节点汇报自己的负载量、剩余存储空间，以及存储的文件块信息等，元数据节点根据搜集到的信息来分配新的存储任务和平衡节点间的存储负载。在多数时间里，客户端读写文件内容都是和存储节点进行交互的，在对文件内容进行操作之前，客户端只需要从元数据节点获取文件名和相应的存储节点，然后直接和存储节点进行数据传输。

最后举例介绍一些其他知名的分布式文件系统。Lustre是一个开源的分布式文件系统，它采用了集群式架构，被广泛应用在高性能计算领域。Frangipani是一个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离来获取极佳的I/O性能。一些互联网公司也根据自己应用的特点，开发了适合自己的大规模分布式文件系统，比如Facebook Haystack，它非常适合存储海量图片文件。

8.4.3 参考实现

上一节介绍了存储即服务几种实现的关键技术。相对而言，SAN和NAS通过较为底层的技术来实现网络化存储，需要专门的硬件设备支持，购买成本和维护成本相对较高，适用于企业内部搭建的私有云中；而分布式文件系统在上层实现了网络化存储，底层可以是普通的硬件设备，适用于具有大规模访问量的公有云。本节将介绍分布式文件系统的的一个参考实现，即Google File System（GFS）。

GFS是Google公司设计的可伸缩的分布式文件系统，用于支持上层应用，比如Google Search和Google Map。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：

（1）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；

（2）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，以应对对超大文件的访问；

（3）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；

（4）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规模地使用。因此，GFS能够很好地支持大规模海量数据处理应用程序。

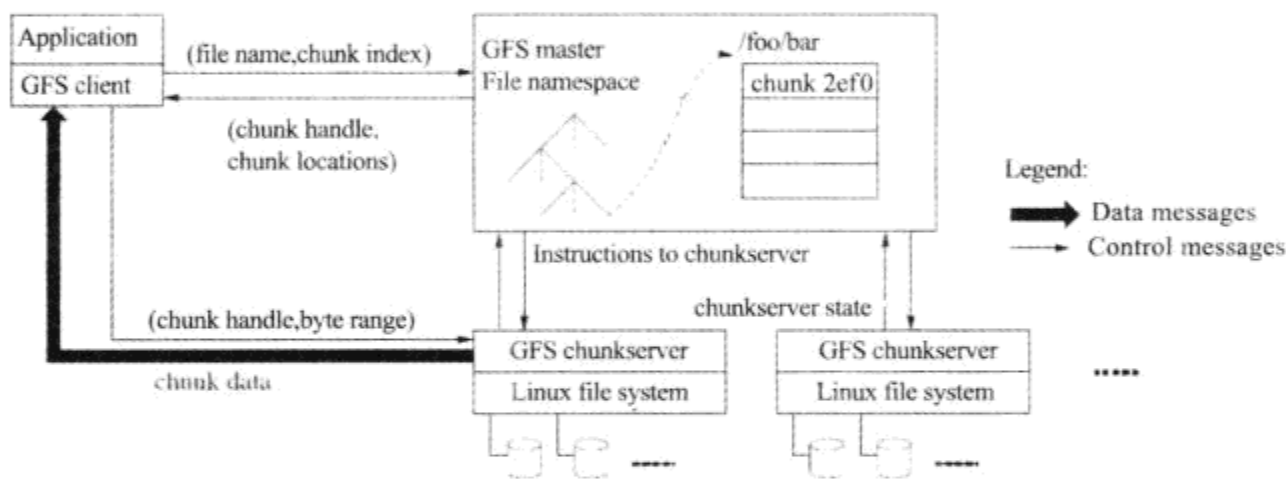


图8.11 GFS架构

如图8.11所示，GFS的实现采用了主从式集群架构——一个GFS主服务器（GFS Master）和若干个GFS文件块服务器（GFS chunk server），GFS主服务器负责管理文件系统的命名空间等元数据信息，GFS文件块服务器负责存储文件块，GFS主服务器通过心跳信息来获得文件块服务器的状态信息，这些信息包括了所存储的文件块列表及存储状态。

在GFS的典型读写场景中，应用通过GFS客户端访问GFS文件系统，GFS客户端根据应用要访问的文件名和偏移量转换成所对应的文件块，并将这些信息发送给GFS主服务器，GFS主服务器在命名空间中找到该文件的信息、相应文件块的句柄及所属文件块服务器，然后将这些信息发送回客户端，客户端获取这些信息以后，直接联系相应的文件块服务器，通过文件块句柄完成读写操作。

从上面的流程可以看出，GFS主服务器只参与了前期的信息交换，而后面的数据交换都发生在客户端和文件块服务器之间。采用这样的设计，即使在大规模数据访问时主

服务器也不会成为性能瓶颈。另外，GFS还采用了“租用”（lease）机制，允许客户端缓存文件块的位置信息，这样在一定时间内客户端都不必再向GFS主服务器交互，大大减轻了主服务器的压力。

2003年Google在著名的国际学术顶级会议SOSP上发表了关于GFS的论文，引起了业界的强烈反响，不少公司或者社区都参照GFS的设计和架构来实现自己的分布式文件系统，其中比较著名的是Apache社区的Hadoop HDFS分布式文件系统，本书后面的章节将对其有所介绍。

8.5 网络即服务

网络资源作为传统IT系统的基础部分为人们所熟悉。网络即服务是网络资源和云计算技术的结合和演变，是对传统网络资源进行的改进和创新以满足云计算服务的需要。本节讨论网络即服务的特点和技术实现。

8.5.1 服务模型及接口

云服务所具有的特点如大规模、高可靠、伸缩灵活、部署快速高效等，对网络也提出了更大的挑战。

首先，在传统网络环境中，网络拓扑往往是固定的或者在很长一段时间固定不变的，这包括网络地址、域名及负载均衡配置等。在云中，这一切都发生了改变。由于虚拟机的失效恢复、迁移及进行资源优化过程中发生的变化，需要对应的网络地址和域名管理服务能够满足动态变化的需求，负载均衡也需要根据网络的变化动态地进行调整，这就需要提供网络相关的服务从而支持对网络的按需调整。

其次，在基础设施云中虚拟机的部署频繁，而虚拟机的大小也达到几百MB甚至若干GB，怎样才能实现快速部署大规模虚拟机实例对网络带宽提出挑战，需要带宽能够按需进行配置。

最后，云作为共享的环境，怎样对不同用户的虚拟机实例进行有效隔离也是一个要解决的问题：用户的实例是自动创建的，所以隔离机制要能够自动化；虚拟机实例是动态变化的，比如根据负载的变化，虚拟实例的加入和删除在不断地发生着，怎样在动态变化的拓扑下对数据和业务进行隔离尤为重要。

由云计算的概念可知网络即服务是将网络资源以服务的方式提供给用户。常见的网络资

源包括路由器、交换机、负载均衡设备，以及其他为云平台定制的网络设备及管理系统。实现云计算的重要基础就是实现虚拟化，对于网络也不例外，通过对网络资源的虚拟化，能够屏蔽底层硬件设备的差异，支持灵活的资源部署和配置，并能够根据业务的需求按需调整。

网络即服务是基础设施层的重要一环。网络即服务为云中的业务提供了多种多样的服务，包括网络地址和域名管理服务、弹性灵活动态的负载均衡服务，以及对网络虚拟化的VLAN和VPN服务等。常见的网络地址和域名管理服务负责的IP地址分配、调整、回收、寻址以及域名解析服务。弹性的负载均衡服务可以动态按需地在用户的多个计算节点之间实现对业务的请求进行动态调度。VLAN技术可以将企业网络划分为虚拟网络VLAN网段，简化网络管理和增强网络安全，是对局域网实现了虚拟化，允许不同组织的用户共享同一个物理的网络。对于跨公用网的安全问题由VPN提供了保障，VPN在混合云中比较常见。图8.12展示了主要的网络即服务类型及与其他形式云服务的关系。

除此之外，安全问题是用户采用云计算最大的顾虑，网络安全在基础设施层是至关重要的。云中的安全与传统的网络安全的目的是相同的，即通过系统的硬件、软件方法保障云中的应用和数据不受偶然的或者恶意的破坏、更改、泄露，保证系统连续可靠地运行，网络服务不中断。

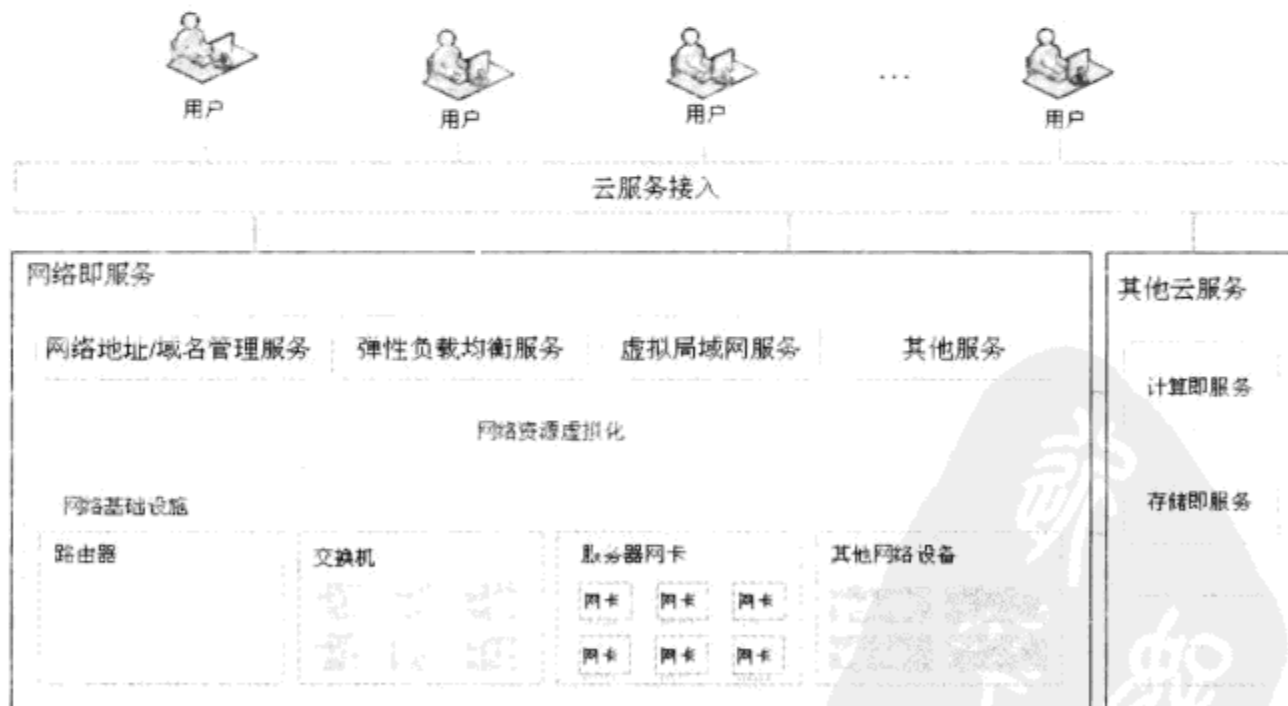


图8.12 网络即服务架构

以基础的网络服务为例，即网络地址的创建、分配、调整及回收，接口模型主要包括以下属性：用于标识网络的显示名称，描述信息，地址空间，地址分配状况，网络的当前

状态等。而网络地址则是基础设施层进行资源分配的基本对象，需要包括地址的名称、状态、描述、子网及绑定的虚拟机实例等。表8.7列举了和网络地址管理相关的基本接口。

表8.7 网络即服务接口

方法	URI	说明
GET	/addresses/	查询用户所有的网络地址列表
GET	/addresses/addl001	查询id为addl001的网络地址的详细信息
PUT	/addresses/addl018	更新id为vol09921的网络地址信息。如安全策略、分组等
POST	/addresses	创建新的地址
DELETE	/addresses/addl015	释放地址addl015

8.5.2 关键技术

下面将主要介绍地址和域名管理服务、弹性负载均衡服务及虚拟局域网服务。

8.5.2.1 地址和域名管理服务

地址和域名管理是网络即服务中最基础的服务，如果没有这两项服务，云中的其他服务只能成为信息孤岛而不能被用户访问。地址服务提供的主要是IP地址的申请、动态绑定和释放等。域名服务为云应用提供高可靠、高效、灵活的域名解析服务，如Amazon的云服务 Amazon Route 53。

在云中，传统的服务正在慢慢变化，对IP的管理要求能够更加灵活。因为在大规模的云环境中，实例的故障甚至是宕机都是普遍的现象，如何保证服务的不间断或是将宕机时间最小化是非常有挑战的问题，需要有故障检测、恢复等机制。这里主要介绍同网络相关的机制，即当一个实例失效时怎样快速启动新的实例并配置完毕，新实例需要沿用失效节点的IP地址。在现有的公有云的实现中甚至每次实例的重启都会引起实例地址的改变。这对IP的分配和管理提出了挑战，因为IP不再确定和某台机器进行绑定，从而需要更灵活的地址管理方式。Amazon提供弹性IP服务，通过IP地址和用户账户绑定，从而由用户根据需要进行实例和地址的绑定。弹性IP服务是网络地址服务的典型代表。

8.5.2.2 弹性负载均衡服务

负载均衡采用某种策略（如轮循策略）在不同的设备（计算机、网络线路等）之间进行负载的分发。负载均衡可以优化资源的使用，保证用户请求的响应时间，提高业务的吞吐

量。负载均衡使得计算能力的扩展变得更加容易，增加一个新的节点，只需要添加设备，并在负载均衡模块上进行配置即可，提高了应用的扩展性。通常采用多个设备进行业务处理，这样可以增加系统的容错能力，提高应用的可用性。负载均衡可以通过程序或硬件来实现。

负载均衡有两种不同的应用场景，一种是为了提高业务的并发能力而将用户的请求按指定策略分发到不同的计算单元并独立返回给用户响应结果；另外一种是为了并行处理将同一个计算任务分解后分发到不同的设备上进行处理，然后将返回结果进行汇总并返回。本节主要介绍前者，即解决并发请求的负载均衡。

前面提到云具有动态性、灵活性和可扩展性的特点，并且需要保障高可用性，这些要求和负载均衡密切相关。在云中，为了提高业务的可用性，通常生产业务都有两个或者更多的计算实例来支撑，在不同的实例之间进行业务分发就需要负载均衡的功能。在业务运行的阶段，随着负载的变化或者节点的实效，需要负载均衡服务能适应实例动态变化。在云中这一切都需要自动完成，所以需要服务的接口提供给业务模块来访问。负载均衡通过和云计算技术结合形成弹性负载均衡服务，现在比较成熟的产品有Amazon提供的ELB服务。

在基础设施层中，弹性负载均衡服务的操作主要包括弹性负载均衡器的创建、配置和删除等。负载均衡器负责在一组虚拟机实例之间进行请求的分发，其中虚拟机实例可以是在同一个数据中心，或者为了提高可用性而分布在不同的数据中心。请求分发的策略也是多种多样的，常见的有轮循策略、加权轮循策略、处理能力均衡、响应时间均衡等。负载均衡器的创建需要用户提供调度策略、虚拟机实例信息、监听的端口及业务的配置信息。负载均衡器应该快速检测失效的虚拟机实例，并且停止向这个节点转发用户请求。负载均衡器可以用Ping来定期测试虚拟机实例的连通性，或者获取业务的可用性信息及反映负载的资源信息等。

伸缩性是云的一个重要特性，在业务的生命周期中，需要根据负载的变化动态调整支撑业务的实例数，会不时添加新的实例或者释放一些利用率较低的实例。弹性负载均衡器提供的动态配置服务支持对负载均衡器的动态调整，从而在负载增加时，将负载分发到新添加的实例上，减轻已有节点的负载；在业务负载低时，可以停止某些实例，负载均衡器也停止向这些实例转发请求，以提高资源的利用率。

伸缩性对负载均衡器更新的效率提出了更高的要求，在负载均衡器的实现上需要考虑这个特点。由于DNS的更新较慢，可能需要数十小时，传统的基于DNS的负载均衡器就不太适合云环境伸缩性要求。Session的处理也是弹性负载均衡服务要解决的问题，如Amazon的弹性负载均衡器就支持本身或者应用生成Cookie并进行管理，这样可以使指定的实例持

续为同一用户提供服务。

8.5.2.3 虚拟网络技术

网络虚拟化的目的是为了屏蔽底层网络基础设施的差异，简化网络资源的管理，对网络数据进行隔离，提高资源的使用率，从而使得多个用户可以共享底层的网络资源，同时保障网络服务的可靠性和安全性。现在常见的虚拟化技术有针对局域网虚拟化的VLAN技术，以及在公共网络设施上构建专用网的VPN技术。随着云计算技术的发展，也出现了虚拟专用数据中心（VPC）技术，这是云计算技术与VLAN和VPN技术的结合，典型代表就是Amazon提供的VPC服务。

VLAN（Virtual Local Area Network）即虚拟局域网，是在云计算出现之前已经存在的技术，最初的目的是为了解决局域网中数据广播的问题，即隔离广播，主要是因为交换机并不能够隔离广播域。通过将物理的局域网继续抽象和划分，能够创建多个虚拟的局域网，不仅能对广播进行隔离，由于不同的VLAN之间不能够直接进行通信从而保证了每个虚拟局域网的带宽和安全。在云服务中引入VLAN可以使网络管理变得简单、方便、灵活、高效和安全，同时降低了管理的成本。

既然VLAN是对物理网络的逻辑划分，划分方法非常重要。目前常见的VLAN划分方法有：基于端口划分、基于MAC地址划分、基于网络层划分和基于策略划分等。不同的划分方式有不同的特点和优势。基于端口的划分是通过在交换机上的配置完成的，根据所连接交换机的端口进行划分。这种方法的好处是简单，但受到交换机的限制，VLAN不能跨多个交换机。基于MAC地址划分比基于端口更加灵活，可以跨不同的交换机而不需要重新配置，在设备位置变化时更方便高效，不足之处就是初始化比较麻烦。采用网络层划分也是比较灵活的方式，如通过IP划分。这种方式可以跨物理网络，配置灵活，管理方便，问题是效率低，在转发时需要对每个网络层数据包进行检查。基于策略的划分方式能够跟业务或者部门结合，比如可以按照公司不同部门的职责对网络进行划分，这种方式更加直观，可扩展性好。目前常见的VPN标准有802.10 VLAN、802.1Q和Cisco ISL标签。

随着Internet的发展和日益普及，一些大中型企业需要一种安全可靠的网络，支持企业的员工、分支机构或者授权的第三方合作伙伴通过公共的网络设施（如互联网）访问公司内部网络。为了解决这个问题，一种方式就是构建专门的网络，包括线路、交换机等，这种实现成本太高，实现复杂，现实中很少采用；另外一种方式就是在公用的网络设施上

建立一个安全隧道，通过对传输的数据进行加密，从而保证数据的安全性，这种实现成本低，管理简单，易于实现，现实中被广泛采用。因为后者不需要专门购置物理网络设施，因而被称为虚拟专用网络，即VPN。图8.13显示的是在多个分支机构之间通过安全隧道连接到总部的虚拟专用网络示意图。

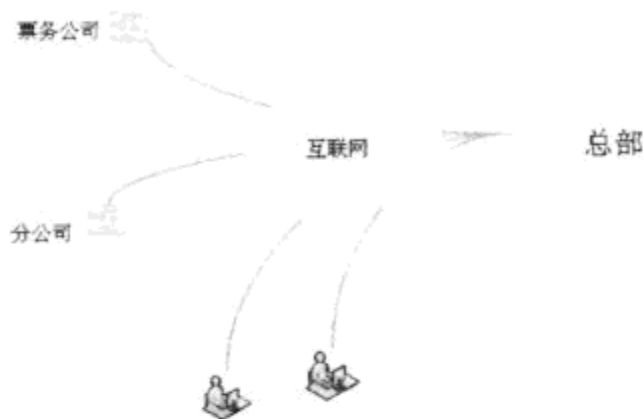


图8.13 VPN网络

VPN是在公共网络上构建安全的、可靠的、低成本的虚拟专用网络线路，它涉及的技术包括隧道技术、加密解密技术、密钥管理技术及认证技术等。隧道如同构建在公网上的专用网络通道，而加密保证了隧道中传播数据不受破坏或窃取，加密解密密钥的可靠管理为数据加密提供了保证，认证技术保证隧道的安全建立。

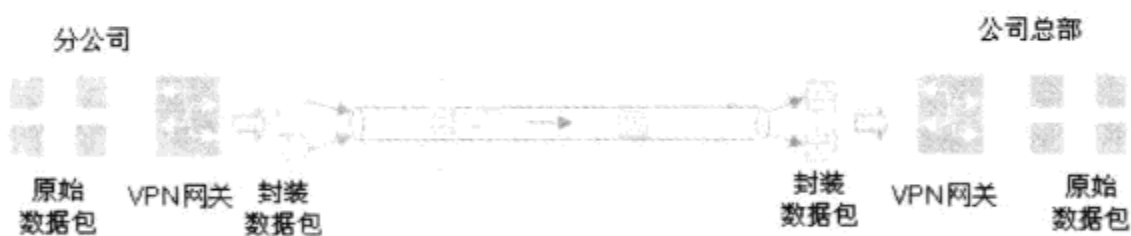


图8.14 VPN工作原理

图8.14介绍了VPN工作的基本原理。这个示例通过在分公司的VPN网关和公司总部的VPN网关进行配置实现，完成认证后，虚拟的专线就建立完成了。“专线”建立之后分公司和总部就如同在同一个局域网中，如果有数据包要从分公司发到公司总部（反之亦然），数据包首先被分公司的VPN网关接收，然后进行数据加密，防止未授权的访问或者篡改，处理后再添加网络报文信息就可以通过互联网发送到总公司。总公司的VPN网关接收到加密数据后，首先进行真实性和完整性验证，确保是授权的访问并且未被第三方破坏，检验完成后便进行解包、解密，然后将原始数据包发送给数据的接收端。

当前VPN的协议主要有四类：点对点隧道协议（PPTP）、第二层隧道协议（L2TP）、在第三层即网络层的IPSec VPN，以及应用层SSL VPN等。下面以IPSec为例进行介绍。

IPSec VPN是基于IPSec协议实现的VPN形式，解决了传统TCP/IP协议的安全问题。IPSec是实现网络数据安全的一整套协议结构，包括用于认证的协议AH（Authentication Header），对数据进行封装的ESP（Encapsulating Security Payload）协议、支持通过互联网进行密钥交换的IKE（Internet Key Exchange）协议及加密算法等。

为了保证数据的真实性、完整性和私密性，IPSec通过AH协议对数据的发送者进行认证，防止未授权用户的访问，通过MD5等协议对数据完整性进行验证，防止数据遭受第三方的篡改，通过ESP协议避免第三方非法窃听信息。ESP协议首先是采用DES、AES等加密协议对数据进行加密，然后进行封装，添加到IP数据包中。除此之外ESP也会提供AH协议的功能，但认证服务的安全性不及AH。AH协议和ESP协议可以同时使用。

基于IPSec协议的VPN有两种不同的工作模式：隧道模式和传输模式。它们的区别在于包的封装格式不同，前者会创建一个新的IP数据包对数据进行封装，而后者则不会创建新的IP数据包。进行数据传输要建立隧道，即在两个VPN设备（网管或者软件）之间建立安全关联，确定认证协议、加密协议及封装模式等的选择。传输隧道建立后就可以通过隧道进行安全可靠的数据传输。还有一个很重要的问题我们没有讨论，即如何选择加密协议，怎样安全管理密钥。这是IKE协议要解决的问题，IKE协议支持密钥交换及建立安全网关。

目前，VPN广泛应用在下面三种场景：用于远程用户访问公司内网的远程访问VPN、连接企业内部各地分支机构的企业内部VPN和同第三方共享网络服务的企业扩展虚拟网（Extranet VPN）。

8.6 小结

基础设施即服务提供是虚拟化的计算资源、存储资源和网络资源，因此。基础设施即服务的种类可以分为计算即服务、存储即服务和网络即服务。本章首先介绍了提供基础设施即服务的服务模型与接口，然后深入介绍了计算即服务、存储即服务和网络即服务。对于每一类具体的服务形式，本章各用一节介绍了对应的服务模型和访问接口、实现服务的关键技术，针对计算即服务和存储即服务提供了一个实现实例以加深读者的理解。

第 9 章

平台即服务

- 9.1 概述
- 9.2 核心系统
- 9.3 扩展系统
- 9.4 参考实现
- 9.5 小结



云计算系统架构的平台层是为应用服务提供开发、运行和管控环境，即中间件功能的层次。基础设施层所要解决的是IT资源的虚拟化和自动化管理问题，而平台层需要解决的问题是为某一类应用提供一致、易用而且自动的运行管理平台及相关的通用服务。在云计算系统架构中平台层位于基础设施层与应用层之间，为应用提供共享的、按需使用的服务和能力。

平台层的功能以服务的形式提供给用户，可以作为应用开发测试和运行管理的环境，即平台即服务（Platform as a Service, PaaS）。平台即服务是云计算平台层的外在表现形式，是云计算平台提供的一类重要的功能集合。本章将首先阐述平台即服务设计所满足的基本需求和所支持的基本应用类型，从而对PaaS的功能特征有清晰了解；然后介绍实现PaaS系统的重要技术，它们是针对PaaS的功能和使用方式而专门设计的。当前被广泛认可的云平台服务主要针对两类应用形式，一类是Web服务，通过快速的请求/响应方式进行交互（称为事务处理类），另一类是数据分析服务，通常处理大量的数据，需要较长的处理时间和巨大的数据空间（称为数据分析类）。最后，本章将通过实例来剖析面向这两类应用的PaaS实现要点。

9.1 概述

与Internet的TCP/IP架构相类比，云平台层类似于网络和传输控制层的地位。即在多层的架构中，需要一个层次来屏蔽下层物理设备的多样性，又要支持上层应用的多样性。平台层即通过一系列的面向应用需求的基本服务和功能来提供应用运行管理的基础，而它本身也屏蔽了基础设施层的多样性，可运行在多样的基础设施层之上。对于大规模不断演化的系统而言，这样的架构选择具有普遍性，它既提供了一定的稳定性从而成为支持演化的内核，又通过模块和层次化为各个子模块的独立演化提供了支持。

9.1.1 驱动力

平台即服务作为云计算中的一类重要服务类型，它的出现有其深刻的商业和技术背景。从商业的角度讲，随着各类网络应用和服务的发展，其规模和复杂度都有了质的变化。这使得应用的快速开发、部署及管理的简化和自动化成为一件必要的事情。从技术的

角度讲，计算基础设施的发展和虚拟化技术的广泛应用使得集中式的、统一的应用运行和管理平台成为可能。

在过去的10年里，运行在企业内部和互联网上的应用，无论是复杂度、类型还是规模都呈现出显著的增长。随着硬件技术的进步，IT用户的业务运作方式也有了很大的变化，企业愈发依赖高效的IT系统和各类应用来支撑其业务。为此，企业必须寻求新的IT系统及应用运作方式，以降低在软硬件的采购和维护、管理方面的大量成本投入。同时，随着企业业务的发展和 innovation，快速开发和上线新业务和应用也是保证企业竞争力的重要手段。企业应用的多样性和灵活性，以及对于应用的高可用性和可靠性的要求，促使企业寻求新的应用开发和运行方式。平台即服务给企业在这方面的需求带来了高效的解决方案。

在云计算的架构中，基础设施层通过基于共享和虚拟化的服务来提供计算能力、存储能力和网络能力。共享和服务的概念也可以适用到应用上，也就是为应用的开发、运行和管理提供统一的环境和平台，这类似于将中间件的概念推广到了云计算的架构中。回顾分布式计算的发展历程，我们看到中间件技术的出现，主要是为了将分布式应用所共同面对的诸如事务管理、资源管理、多线程及进程间通信等通用性的问题及其具体实现抽象为简单应用编程接口或系统服务，以简化应用的开发和管理。

同样，在云计算的大框架下，平台即服务提供了进一步的抽象。我们通过对云应用进行分类，总结相关实践中的共性问题，抽象出特定的模式和解决方案。这些共性的部分，包括负载均衡、缓存、半结构化的数据存取、大规模的消息通信等。将这些共性的部分剥离出来，由专业的人士进行开发，通过服务的方式提供给应用使用且由专业人士维护，这样使得它们的功能、性能和成本都较企业自己独立完成更好、更省。因此，在中间件的基础上进一步简化应用的开发和管理而提供一组服务，即平台即服务，可以提高应用的灵活性，降低运行管理的开销。

PaaS包括一系列的平台软件和基本服务。PaaS的提供商在平台软件 and 基础服务的实现上具有多样性，各自针对用户对平台的一类或几类特定需求和使用方式。根据所针对的应用类型，PaaS在理念、客户定位和实现方式上也会存在差异。比如，GAE (Google App Engine)就预先定义了所支持应用的基本模型，以应用编程接口 (API) 的方式与应用进行交互，应用通过调用相应的编程接口来调用平台层的功能，从而实现应用的功能和运行管理。

通常的企业客户维护着大量现存的应用。企业可以考虑把这些应用迁移到云计算平台层上，或者保持维护现有应用，而把新的应用运行在云计算的平台层上。对于前一种选择，企业需要考虑向云计算平台进行迁移转型的成本；而对于后一种选择，企业需要考虑做到两类系统的有效集成。云计算平台的提供商需要考虑如何有效地平衡用户的这两个方面的需求。这两类需求将影响到平台层的功能、实现方式和推广模式。PaaS技术的发展将会更加贴近客户的多样化需求。

9.1.2 主流类型

当我们开发各种各样基于网络的应用时，通常会把这些应用中共有的部分或者都要使用到的功能抽离出来作为基础服务，以供编写和运行应用时调用，从而降低应用创建和运维的复杂性。这一系列应用所要用到的基本功能即为平台层所提供的服务。

当前，PaaS上运行的应用主要分为两类，一类是Web服务，另一类是数据分析服务。前一类主要是通过浏览器访问、采用请求/响应模式进行交互的应用，称为事务处理类应用。事务处理应用的要求主要包括快速响应、高可用性、大并发量等。后一类应用主要是对大量的数据进行分析处理，称为数据分析应用。数据分析应用的主要要求包括强大的计算能力和存储能力，对于实时性的要求不高，数据处理完毕后任务就结束运行了。为了支持这两类应用，PaaS系统根据应用特点而有专门的设计，这将在9.3节有细致的介绍。

根据所针对的应用类型，PaaS通过编程模型和接口与应用进行交互。关于所支持的编程模型，PaaS可以基于标准编程模型，也可以基于自定义编程标准。基于标准编程模型可以降低用户的使用门槛，并且使得已有的应用系统更容易迁移到云平台上，比如在Google App Engine中，可以直接使用J2EE模型进行Web编程。某些PaaS为了更好地解决云计算中的某类特殊问题而采用了自定义的编程模型，比如force.com为了更好地支持多租户技术而自己定义了Apex编程模型。对于数据分析类应用，PaaS通常支持MapReduce模型。

作为支持某种类型应用的通用基础功能的集合，平台即服务的类型及其功能也会随着应用的发展而变化。例如，支持大规模网络游戏的基础平台、支持社交网络的平台，或者面向大规模数据存取操作的半结构化数据存储和非关系型数据查询平台，都可能或者正在发展形成新型的PaaS类型。可以预见，随着市场规模的扩大和市场细分的深化，PaaS的种类及提供PaaS的厂商将会不断增加。这些都会进一步丰富云应用的类型和开发实现方式。

9.1.3 功能角色

从平台层所处的架构层次和所针对的用户需求来看，PaaS的本质是以共享和基础服务的方式来满足多样性应用运行管理过程中的共性需求。因此，从用户需求的不同角度来看，PaaS也展现出不同的面貌特征：从架构层次来看，它是共享的中间件平台；从功能特征来看，它是集成的软件和服务平台；从使用模式来看，它又是虚拟的应用平台。

共享的中间件平台：从架构层次上来看，平台层是为了有效支撑大量应用实例的运行管理，它是一类应用运行所需要的资源和服务集中起来并进行共享的中间件平台。在传统的中间件环境中，考虑应用的性能和可靠性，通常用户需要为每个应用维护一套单独的环境，除了硬件平台之外，还需要有单独的运营团队来进行中间件的选型、部署和配置等。另外，为了保证应用的可靠性及突发的负载，往往需要准备额外的软硬件资源。

如图9.1所示，PaaS将这种传统的静态、独享的中间件平台转变为一种动态、共享的中间件平台。每个应用将在云平台上统一进行管理和运行。平台层既提高了资源的利用率，又通过对应用和平台进行概念和功能的分离进一步简化了应用和平台的运营和管理。

PaaS将“共享”扩展到更大的范围。与基础设施层所共享的对象不同，PaaS所共享的对象是应用运行所需的资源和基础功能。PaaS通过动态资源调度实现了计算资源在不同应用之间的共享和按需供给；通过基础服务如流量平衡器、专门的消息服务机制实现了不同应用之间的基础功能；通过统一的管理平台实现应用运维管理的功能和方法的共享。

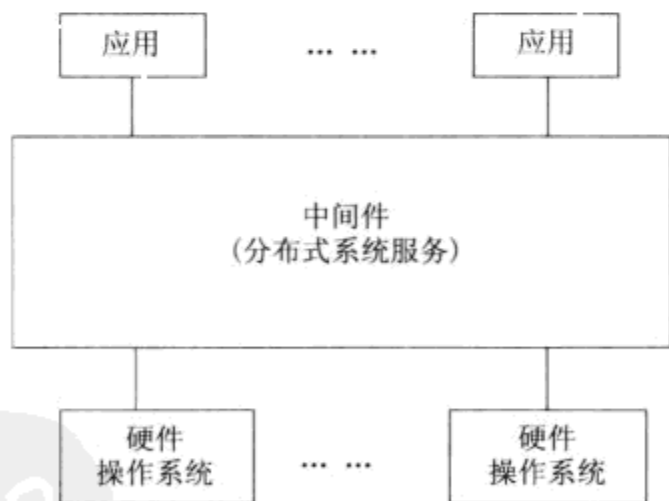


图9.1 PaaS作为扩展的中间件提供应用所需的资源和共享的基础功能

集成的软件和服务平台：从功能特征的角度来看，平台层整合各种不同的软硬件资源向应用提供一致而统一的资源和功能。通过整合，应用运行所需的各种资源和基础功能以

统一的编程模型和调用接口暴露给应用使用，应用无须关注下层的细节。同时，PaaS平台根据所支持的应用类型，可以精心选择和优化所提供的应用的资源和服务，使得应用的开发和运行变得更为简单高效。

如图9.2所示，平台即服务可能建立在多个基础设施服务之上，需要对应用提供一个一致的、单一的基础设施视图。PaaS还需要面向云环境中的应用提供应用在开发、测试和运行过程中所需的基础服务。平台层除了提供Web服务器、应用服务器、消息服务器等传统的中间件以外，还需要提供其他相关的管理支撑服务如应用部署、应用性能管理、使用计量和计费。另外，云应用本身可能就集成来自不同云服务提供商所提供的功能或服务，这些也需要平台层提供相关的支持。

比如，一个企业可能将自己的应用运行在企业内部所建设的PaaS上，将客户信息保存在企业内部的数据库里，将一些非敏感信息如产品手册和图片等文件直接放到Amazon S3（简单存储服务）中以节省存储服务器的采购成本；为了方便与客户交流，该应用甚至可能直接集成Microsoft的Live服务。为了支持应用这些功能需求，PaaS应该提供一致的访问接口和编程模型，从而使得应用通过简单的接口调用就获得相应的功能，而无须单独与各自的服务分布打交道。例如，上层应用通过PaaS所提供的统一接口来对本PaaS内部的数据和存储在S3中的数据进行透明访问。

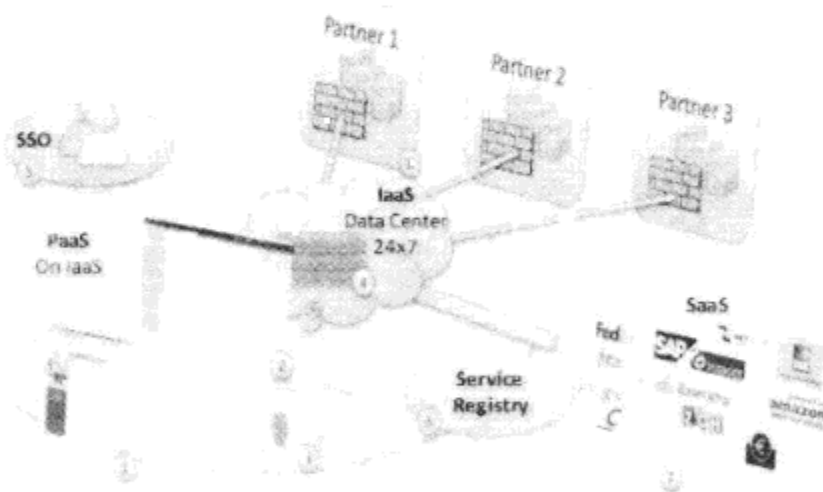


图9.2 位于IaaS之上的PaaS面向应用整合了各种软硬件资源

虚拟的应用平台：从使用模式上看来，作为应用运行管理的环境，PaaS模糊了物理资源的限制，在应用看来是一个按需索取、无限可扩展的虚拟平台。PaaS作为云应用的运行环境，云应用通过PaaS所提供的编程接口（API）按需获取运行所需要的各种（虚拟的）资源和能力。一般来讲，资源的获取是动态即时的。例如，平台层根据应用程序的负载起伏，动态估计所需的计算和存储资源，按照服务质量的约定（SLA）按需提供所需资源。

从自动化的角度来看，PaaS的基本目标是使应用更加专注在用户的功能性需求上，而平台则自动为应用满足诸如负载均衡、自动规模调整等非功能性的需求及管理的需要。

在传统的应用开发中，用户需要花费大量时间和工作来进行软件栈（即中间件）的选型、定制和部署。在PaaS上，这部分工作将由PaaS根据应用的需求和特点自动完成。我们可以看到，在Google App Engine和force.com这样的PaaS上，用户不再需要手动地选择自己所需要的Web容器和数据库产品及选择和扩展所需的管理功能。用户更多地是针对应用的特点，对诸如运行时服务质量（QoS）需求、伸缩策略或者部署方式等应用的参数进行指定和配置，平台层可以根据应用的相关配置自动提供支撑应用的软硬件资源，以及在运行时进行自动的负载均衡、伸缩控制和SLA优化等。

总之，PaaS针对的是有效而自动管理大量应用的需求。PaaS的功能和结构设计要满足这样的需求。下面我们将从功能和技术的角度来看平台层如何满足这样的需求。

9.2 核心系统

一个平台一般来讲也只能对某类应用进行高效、方便的支持。在当前的IT实践中，存在大量不同类型的应用，因此，从用户的观点来看，PaaS与PaaS之间也存在巨大的不同。然而，在实现PaaS的过程中，我们也会发现，PaaS作为一个系统，其中的功能和模块大致分为两类，首先是PaaS的核心系统，包含了PaaS的一系列本质特征，即使是在不同的PaaS中，也会有这些特征的实现；其次是PaaS的扩展系统，主要包含了针对其支持的应用类型的支持，比如GAE作为支持事务型Web应用的PaaS，就包含了数据访问和缓存的相应模块。本节就讨论通用的核心系统。

9.2.1 简化的应用开发和部署模型

对于一个IT应用的要求，一般分为两个方面：一方面是与业务相关的功能性需求；另一方面就是诸如安全性、可靠性及服务质量等非功能性的需求。应用的开发阶段主要考虑功能性要求，而运行阶段主要关注非功能要求。不同的应用在非功能性要求方面具有一定的相似性，为了支持这些非功能性要求，人们通常总结出一定的功能模块和模式。比如，

在不同Web 2.0应用的高性能方案中，我们一般都能发现诸如负载均衡、反向代理及数据缓存等相似模块。这些模块和模式是平台层支持应用运行的基本方式。平台层的一个重要目标就是把业界在过去多年来在分布式应用中获得的经验总结起来作为服务提供给用户，使用户能够将更多的精力放到与业务相关的功能性需求上去。

前面提到，平台层的基本目的是为了进一步简化大量应用的开发、部署和运行管理。在平台层，为了实现简化应用开发和部署的目的，应用一般被定义为功能性模块和一系列策略的组合。在进行应用开发的时候，开发人员只需考虑业务功能的实现；而非功能性要求通过选择所提供的策略配置来表达。平台层在应用具体部署时根据这些策略选择自动提供相应的资源、服务功能及其配置。

我们可以通过一个关于数据高可用性的例子（如图9.3所示）来看平台层给用户所带来的便利。假设用户A和用户B都需要CRM的应用，用户A需要很高的数据可用性，而用户B则不太关心这个问题。开发人员只需要进行一次开发就能满足这两个企业的功能性需求，而应用管理人员则只需要在部署的时候根据业务的需求选择配置策略，云平台会自动为它们产生不同的部署和配置。比如，在本例中，用户A的高可用性需求通过主从方式的商业数据库来满足，而用户B的数据可用性配置为开源数据库的定期备份。

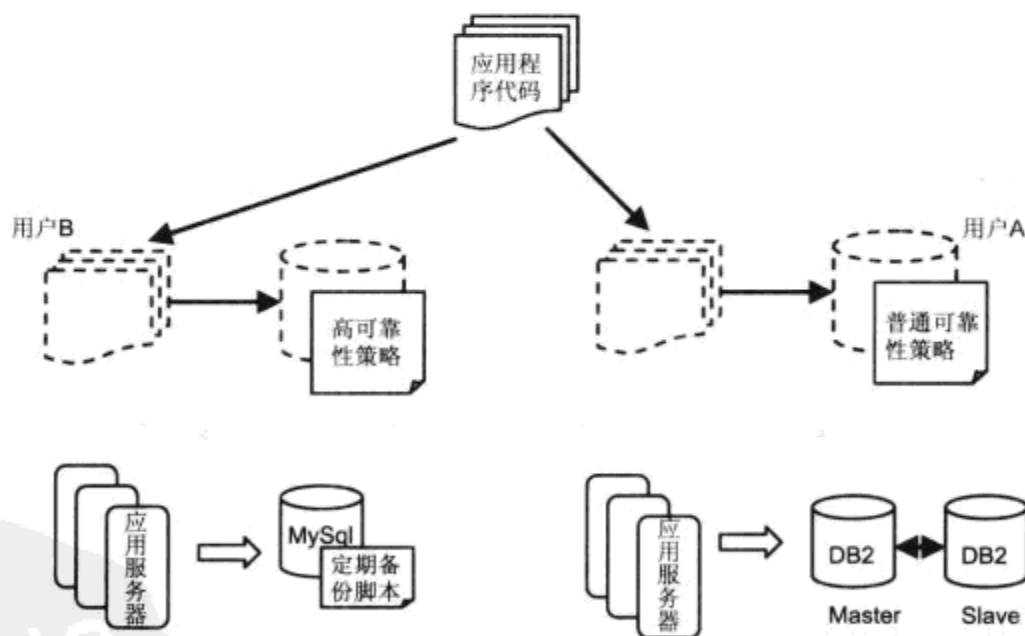


图9.3 云平台中，应用的开发和部署体现了功能性和非功能性要求的分离

一般来说，传统的中间件已经有功能性和非功能性要求满足的分离。在云计算平台层，这种分离变得更为彻底。例如，应用管理无须考虑应用的容量需求而仅需配置所需的服务性能策略；应用所需资源是自动按需供给的，而无须预先准备好；应用所需策略的实

现可以随着时间推移而不断改进，但应用无须感知策略实现的内部变化。总之，这种分离使得应用的开发更加简单，而应用的运行更为自动化。

9.2.2 自动的资源获取和应用激活

为支持应用的运行，平台层需要为应用分配相应的资源，包括计算资源、网络资源和存储资源。建立在基础设施层上的平台层可以通过调用基础设施层的功能和接口获得相应的资源并分配给相应的应用，也可以直接实现基础设施管理的功能，而无须抽象出单独的基础设施层。平台层对应用所需资源的管理分为两个方面：在应用部署上线的时候所需初始资源的分配，以及应用运行过程中根据性能要求进行动态的资源调整。两者所采用的方式有所区别。前一种方式根据应用的初始配置元数据而决定资源的种类和数量；后一种方式根据应用的运行负载变动和性能目标采用动态的模型计算所需的资源种类和数量。

前面提到，提交给云平台的应用以功能模块加上非功能性策略进行定义。云平台层首先根据应用的这两方面的需求计算出支持这个应用所需要的资源类型、配置模式和相应的数量，比如：虚拟机数量及其CPU、内存配置，以及所需要的各类中间件、功能软件、网络连接和存储空间等。简单来说就是为应用创建虚拟的服务器，包括服务器上的软件栈和相应的配置。

当为应用配置好运行所需要的各类资源之后，平台层还需要进行激活操作，让应用确实运行起来并正常提供所应有的功能。应用的激活涉及所分配基础设施层的资源的配置和激活、平台层的功能和服务的配置与激活，以及应用层的功能和配置的启动。平台层所做的激活工作主要包括：

首先需要完成的是资源之间的依赖关系的解析和支持。在定义应用的时候，不会也不需要考虑支持应用的下层资源和基础服务的关系和结构，以及提供这些资源和服务实例的具体配置。应用所涉及的各功能模块之间一般通过一系列的逻辑关系来进行连接。平台层在激活应用的时候需要将应用的各功能模块之间的这种逻辑连接与具体的实现实例的细节和配置相关联。比如，平台层上的应用在定义的时候指定功能模块A与功能模块B分别访问不同的数据库，以逻辑名数据源A和数据源B来进行标示。在资源分配过程中，平台层将数据源A和数据源B分别实例化为不同的虚拟机和数据库。在应用激活的过程中，平台层针对功能模块A和B分别进行配置，应用服务器与数据库服务器之间将使用不同的数据连接实例，使用不同的数据源IP和名字等。

在应用的部署过程中，可能涉及应用模型没有指定但实例化的时候必须存在的功能模块，如一个应用服务器从单个变为多个时需要在前端添加负载均衡器，或者数据高可用性配置中需要多个数据库实例并配置为主从关系。这些根据应用部署情景而即时添加的模块是应用激活所需关注的。

除了解析并实现资源之间的依赖关系外，在应用能够正式运行之前，还需要对资源进行一系列的配置和初始化工作。一般来讲，在资源分配阶段所建立的软件栈中，各软件资源都只包含默认的配置。而不同的应用则需要对这些软件进行配置的更改，比如针对应用的不同性能需求配置不同大小的连接池，或者指定不同的缓存失效时间等。另外，平台层还需要对某些软件资源进行一系列的初始化工作，比如，在数据库服务器实例上为应用创建相应的数据库和表格，或者从其他用户指定的数据源导入数据等。

平台层的一个基本特点是资源分配和应用激活的自动化操作。应用管理员在提交对应的应用并指明所需的配置策略后，平台层能够自动解析应用的配置，转化为对应的资源分配和配置选项，并且通过调用基础设施层的接口实现资源分配，通过其自身或者应用所提供的配置工具实现应用从（虚拟）服务器到中间件直到应用自身的配置和启动。

9.2.3 自动的应用运行管理

在传统的中间件环境中，需要专门的应用运营管理团队针对应用进行部署、配置和运维管理。具体的实施则是一个比较长的过程，需要选择数据中心，规划网络连接，购买服务器，安装操作系统和中间件，根据应用的需求进行配置等。如果应用的需求发生了变化，无论是应用升级还是根据性能要求等非功能性需求更改应用的部署和配置，都需要有一个繁复的计划和实施过程。由于这个过程涉及软硬件多个层次的内容，因此需要具有软硬件环境的大量专业技能和知识。应用的运维管理是企业IT支出中非常重要的一部分，许多企业需要设置专门的团队来负责这项任务。因此，简化应用的运维管理是从应用在线运行角度对PaaS提出的基本要求。

实际上，从应用的角度来讲，企业关注的是应用的功能及应用是否正常运行。为了保障应用的正常运行，企业需要负责管理好应用运行的整个软硬件环境。而PaaS则提供了应用运行环境和应用自身的分离；应用运行所需的资源、基础服务和管理操作都可以交由PaaS平台来负责，甚至通过一系列技术实现自动化操作。在PaaS平台上，企业仅需关心应

用的功能，监控应用的运行是否达到要求，而无须关心如何准备好各种资源和条件来使运行正常运行，后者由PaaS平台来照管。

为了实现以应用为中心的管理模式，PaaS展现给用户的是以应用为中心的逻辑视图，即展示应用逻辑层次上的功能模块，以及通过属性和策略所表达的非功能性约束。用户在应用的逻辑视图上进行应用的管理，比如监视应用的性能、更改应用的属性和策略等。PaaS平台则动态地调整应用所需的资源和管理策略，以保证达到应用管理人员设定的对应用的性能及其他方面的要求。以应用为中心的管理方式是PaaS简化应用管理的基本思路。我们在上一节就已经看到，用户的非功能性需求通过逻辑性的策略表达，消除了用户对特定中间件的技能需求，也进一步简化了应用的部署和管理工作，降低了成本。

可以通过对于应用运行时的可靠性和伸缩性进行比较来体现PaaS相较于传统的中间件环境中管理方式的优势。在传统的中间件环境中，为了保证可靠性和伸缩性，除了需要在部署阶段进行支持（比如，数据库的主从配置），也需要在运行时付出大量人力。一般来讲，在传统的中间件环境中，应用的运行状态，比如中间件实例的数量，每个实例的具体配置，对于管理员来说都是可见的。管理员需要随时监视应用（及中间件）的运行状态，人工判断是否存在节点失效或者负载过高等情况，一旦异常发生，管理员根据事先制定好的工作流程来启动备用的服务器，运行相应的管理脚本来对新的服务器进行配置和初始化等。在这个过程中，一般都会对应用产生或多或少的影响，比如系统的扩容一般会停机维护。

在PaaS平台中，这一系列工作则由PaaS自动完成，用户在部署应用的时候只需要指定相应的可靠性和自动伸缩策略，PaaS平台会自动跟踪和管理应用的运行状态，并调整资源供给来满足策略要求。比如，当Web服务器负载过高时，PaaS平台会从基础设施层申请新的虚拟机运行Web服务器形成新的实例，根据应用的属性进行相应的配置等。这些动态的变化在应用的逻辑视图上对管理员来说可以是完全透明的。

除了向应用管理人员提供应用的逻辑视图，PaaS还通过在逻辑视图上加载适当的管理操作接口从而形成一致的集成管理视图。集成的管理视图既展示了应用的运行信息，又展示了管理人员可以使用的管理操作，使得以应用为中心的管理更为直观而简化。一方面，管理人员可以在同一个管理控制台中获取来自不同层面的管理信息；另一方面，这些应用运行信息通过以应用为中心的方式展现出来，比如，整个应用的并发请求数量或者平均响应时间等，而不是某个具体的中间件实例的负载状况。在传统的中间件环境中，管理员直接使用不同的中间件和系统管理工具，对每个中间件实例及对应的运行环境进行直接的管

理，往往是一个手动使用分散的管理工具及人力介入进行决策的过程。在PaaS平台中，管理人员可以将精力集中在与应用相关的监控和管理上。

实际运行中，PaaS会搜集基础设施层和平台层上原始的各类运行状态信息，比如每个虚拟机的CPU和内存使用状况，或者每个中间件实例的运行状态，每一个应用的访问量和消息服务响应时间等。PaaS对这些原始信息进行一系列的处理，以便实现以应用为中心进行数据的展示，同时根据这些数据和应用的配置策略进行相应的资源调整操作。当PaaS平台运行在不同的基础设施层上时，这些基础设施层的差异也会被PaaS屏蔽，以便展示给应用管理人员一致的应用运行管理视图。

在传统的中间件环境中，策略的实施往往是针对某类资源单独进行，比如网络的配置，或者某类中间件的设置等。PaaS平台将提供更加灵活的策略组合和集成的策略实施。根据应用的运行动态，PaaS可以实时调整提供给应用的各类资源。比如，在高负载的情况下，为保证应用响应时间，一般需要在应用服务器和数据之间配置缓存查询数据，而这类缓存往往会提高应用运行的成本。因此，管理人员可以配置PaaS按照应用性能和当前访问量自动启动或者停止类似的数据缓存服务，以降低应用运行的成本。

9.2.4 平台级优化

在PaaS中，优化在两个层次上进行：在应用层次，针对应用的性能和配置策略，PaaS动态调整应用所使用的资源，在保证达到应用要求的前提下尽量提高资源的使用率，降低应用的运行费用；在PaaS平台层次上，在保证各个应用的运行要求下，PaaS通过资源的共享和复用从而降低平台的运行开销，提高运行效率。可以看出，这两个层次的优化在目标、范围、手段和实施者等方面是各不相同的。

将应用运行在PaaS平台上，应用的所有者不再拥有单独的软硬件平台，针对软硬件平台的优化工作也需要由PaaS来进行。在传统的环境中，往往需要有经验的管理员通过大量的分析和观测工作才能形成合理的优化方案，实施优化也需要大量的工作量和时间。在PaaS平台上，这一系列工作被简化了，PaaS可以自动发现优化的模式，可以通过虚拟化的能力自动、快速实施，而不影响到应用的运行。

同时，PaaS平台执行优化工作时，也需要调用所依赖的层次，如基础设施层的功能和服务来完成。PaaS平台可以根据应用的策略及运行情况来自动进行跨层次（平台层与基础

设施层)的优化和调整。例如, PaaS可以根据应用的不同组件之间的消息传递情况, 将通信量高的组件通过迁移技术逐渐调整到物理上靠近的位置, 甚至同一台物理机器上, 以提高I/O效率。PaaS可以根据基础设施层所提供的资源性能和价格差异, 从而将应用部署或调整到相应的资源上, 以便优化应用的运行性能和成本。

PaaS平台上运行着大量的应用, 由于规模和自动化的要求, 其上的优化也将面临着大量的挑战。这是未来大规模分布式系统研究和实践的重要方面。

9.3 扩展系统

上一节从商业和技术发展的角度来看PaaS的发展动力和所扮演的重要功能。总结来说, PaaS就是面向应用提供集成的、共享的应用开发和运行环境及一组服务。为了支撑大量的应用运行实例, 这样的环境将应用所需的基础功能抽离出来作为基础服务通过编程模型和接口暴露给应用调用。

PaaS环境暴露给应用的基础功能既应满足应用的需要, 又要考虑大量应用实例共享这些基础功能所带来的压力。因此, 合理的基础功能抽象及有效的设计和实现来满足应用对这些基础功能的需求, 将成为一个PaaS平台成功的关键。PaaS平台借鉴了传统中间件中的一系列技术, 如流量平衡、消息通信、缓存管理、容错和错误恢复等, 同时PaaS也创新了一系列新的技术, 如非关系型数据存取、大规模消息通信、自动伸缩机制等, 从而提高应用运行管理的有效性和自动化程度。

针对两类当前广泛支持的PaaS应用, 即事务处理和数据分析, 本节选取几项关键技术来介绍它们如何支持和简化了对应的应用并实现规模要求, 这些技术包括非关系型数据存取、大规模消息通信、海量数据分析等。PaaS采用的技术来自于大规模分布式系统方面的进展, 也推动着这方面技术的发展。希望通过本节的介绍让读者能够进一步从技术角度把握云平台层的特征和能力。

9.3.1 非关系型数据存取

大量应用都涉及通过数据库进行数据的存取。通常采用的数据库是关系型数据库如

Oracle、DB2等。随着网络应用的发展，数据类型也丰富起来，如图片、文本甚至多媒体数据等。这些数据的存取采用需要固定表结构的关系数据库进行读取效率不高。当前，人们已经认识到非关系型（NoSQL）数据存储不需要固定的表结构，通常也不存在连接操作，在大数据存取上具备关系型数据库无法比拟的性能优势。

为了支持多样数据的存储服务的高可用性和可扩展性，非关系型数据库的架构通常比较复杂，而且具体实现也不尽相同。我们分析了几种主流的非关系型数据库，从应用接口、数据管理、节点管理三个方面总结了它们的具体实现中的关键之处。

9.3.1.1 非关系型数据库的应用接口

应用接口是非关系型数据库提供给用户的编程接口，通过该接口用户可以访问非关系型数据库的数据存储服务。相对传统的关系数据库而言，非关系型数据库的应用接口有两个特点：简单和灵活。对于普通用户而言，非关系型数据库只提供几个简单而有限的接口：存储，获取和删除，而且这些接口都是通过键（key）来操作的。所谓灵活，是指在允许的数据大小范围内，用户可以存储任意格式的数据。

在非关系型数据库中，数据操作接口与关系型数据不同。存储操作代替了传统数据库的插入和更改操作，它需要用户至少提供键及所对应的数据项。例如，在Cassandra里，它的存储数据项是以行为粒度进行操作的。Cassandra的存储操作为insert（table, key, rowMutation）。这里的table是键值空间，key是数据项的键，rowMutatable是行粒度的数据项，该行可以包括多个列值。另外，在某些实现中，比如Dynamo，存储操作还会要求提供一些上下文对象，这些上下文对象包含了存储数据项的版本信息和所在节点信息，这些信息对于数据管理非常重要。获取操作通过键及其他相关信息来获得对应的数据项，返回给用户使用。在非关系型数据库中，考虑到应用的特殊需求，删除操作通常并不是物理上删除数据项，而是将该数据项标识为已删除，有些非关系型数据库甚至不提供删除操作。

一般而言，应用接口的通信协议有两种：HTTP和Protocol Buffers。HTTP方式可以承载XML和JSON这些用户可读的数据和信息，而Protocol Buffers将结构数据序列化传输，然后在接收端将其反序列化。由于序列化的数据更小，从而实现快速传输，提高系统效率。这两种方式都是与平台和语言无关的。

9.3.1.2 非关系型数据库的数据管理

通常，非关系型数据库是一个分布式系统，它所存储的数据项分布在各个节点上。

该系统的数据库管理需要保证数据读写的高可用性，其关键技术涵盖了分区、数据复制和读写、数据版本管理等几个方面。

分区是保证非关系型数据库可扩展的基础，它涉及数据和节点的管理。目前，大多数非关系型数据库的分区都是通过一致性哈希来实现的。所谓一致性哈希，是将哈希函数所有的哈希值构成了一个首尾相接的环（最大值和最小值相接），组成非关系型数据库的每个节点在环上随机选择一个哈希值，将该环分割成N份（N为节点数目）。在选取哈希函数时，要尽可能将节点均匀分布在哈希环上。同样地，需要存储的数据项也通过该哈希函数将自己映射到哈希环的某个节点上。于是，负责存储的节点和需要存储的数据项都映射在了同一个哈希环上，并且某个数据项由从它的哈希值出发沿哈希环顺时针方向遇到的第一个节点负责存储。

如图9.4所示，哈希环的取值空间为0~232，组成该系统的4个节点均匀分布在哈希环上，哈希值落在(节点1, 节点2]之间的数据项由节点2负责存储，哈希值落在(节点2, 节点3]之间的数据项由节点3负责，哈希值落在(节点3, 节点4]之间的数据项由节点4负责，哈希值落在(节点4, 节点1]之间的数据项由节点1负责。通过这样的分区方式，数据存储的任务被均匀分摊到各个节点上。

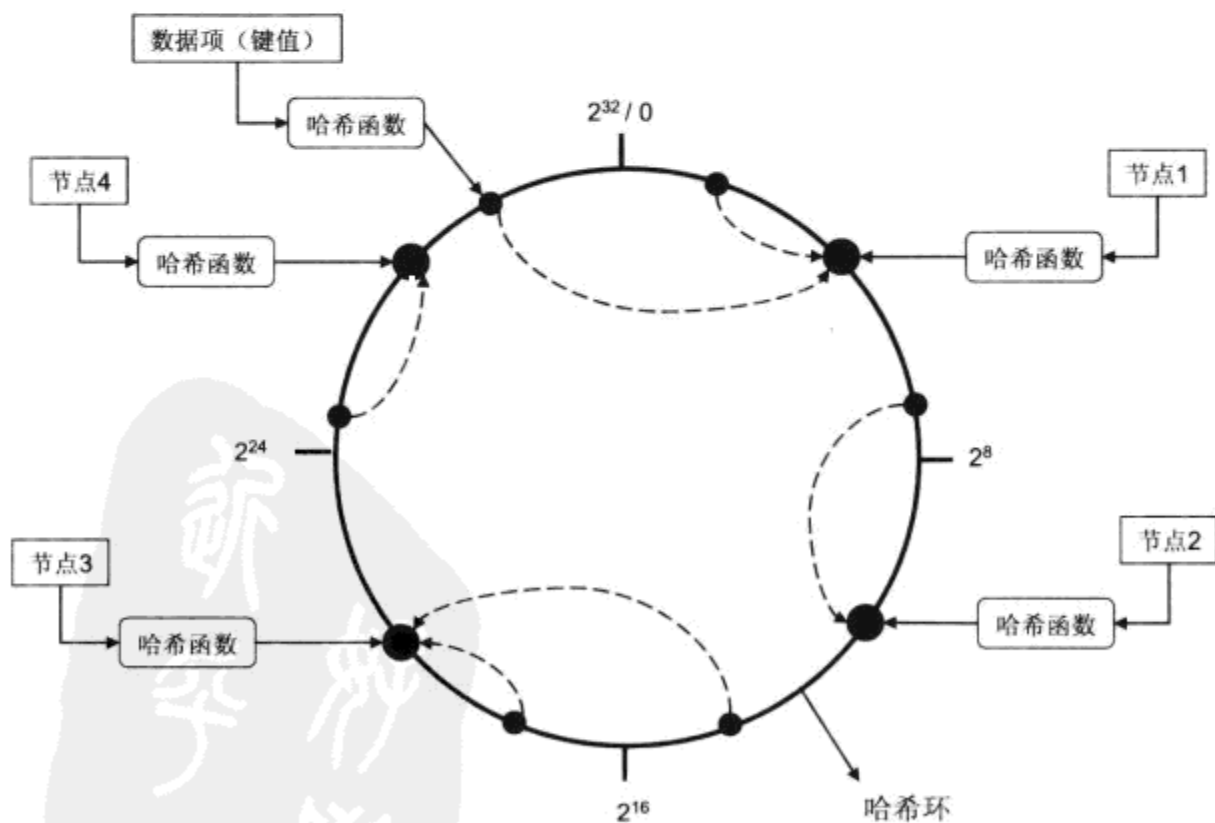


图9.4 采用一致性哈希来管理数据

不过，在实际应用中，哈希函数并不能完全实现均匀分配，从而造成某些范围内的

数据项过多、分区不平衡的情况。目前，有两种主流的方法解决这一问题：增加新节点或采用虚拟节点。增加新节点是将负载过重的范围拆分，让一部分负载分配到新增节点上，Cassandra就是采用的这种方法；采用虚拟节点是将整个哈希环细分成若干小范围，每个虚拟节点负责一个小范围，每个节点负责若干个虚拟节点的存储，对于负载较重的节点，可以将它的部分虚拟节点重新分配到其他负载较轻的节点上，Dynamo和Riak就采用了该方法。总之，分区是实现分布式存储的基础，数据管理和节点管理的其他部分都与它有关联。

数据复制是指通过在多个节点上进行数据冗余备份来提高数据的可用性和可靠性，一般数据复制所要求的节点数 N 为大于等于3。从前面的分区部分我们了解到，某个数据项是由从它的哈希值出发沿哈希环顺时针方向遇到的第一个节点负责存储的，那么其他节点如何选取呢？目前的基本思想是继续沿哈希环顺时针方向选择节点，以达到备份所需的要求。不过，如果所选取的节点都位于同一个机架上，在该机架整体失效的情况下，系统的可靠性将大为削弱。因此，在选择节点的时候还要考虑节点的物理位置，如在不同的机架上，甚至要分布在不同的数据中心。

前面提到，非关系数据库为了提高读写效率，在写数据项时，并不一定需要在 N 个节点上成功写入以后才返回，即 W （成功写入的节点数） $\leq N$ （数据复制的节点数）；同样，在读数据项时，也不需要 N 个节点上成功读取以后才返回，即 R （成功读取的节点数） $\leq N$ 。根据分布式系统的Quorum理论，可以证明在 $W+R > N$ 的情况下，系统能够保证强一致性，而 $W+R \leq N$ 的情况下能够实现最终一致性，也就是说需要牺牲同步时间来换取快速的读或写。

由于每个数据项保存在多个节点上，客户端在每次更新该数据项时，成功写入的节点（ W ）并不一定完全相同，并且小于数据复制的节点数（ N ），这就造成了数据版本不一致的问题。向量时钟（Vector Clock）是解决这个问题的经典方法，也被广泛应用在流行的分布式非关系型数据库中，比如Cassandra，Dynamo和Riak。通过时钟向量，节点给每个数据项标记版本信息，客户端每次更新数据，相应的版本信息就递增，各个节点之间通过版本信息来解决冲突和实现合并，从而达到数据一致性。

如图9.5所示，同一个数据项存储在三个节点上，在 a 时刻节点1上的数据更新了，版本信息标记为（1，0，0）；在 e 时刻节点3上的数据更新了，版本信息标记为（0，0，1）；在 b 时刻，节点1上的数据再次进行了更新，变为（2，0，0）；然后，节点1上更新的数据传播到了节点2上，这个不是由客户端发起的数据更新不改变版本信息；在 c 时刻，节点2

上的数据被客户端更新了，版本信息变为 $(2, 1, 0)$ ；在d时刻再次被更新，变为 $(2, 2, 0)$ ；接着，节点2上更新的数据传播到了节点3上，这时，节点3上有两个版本 $(0, 0, 1)$ 和 $(2, 2, 0)$ ，需要由上层应用或既定规则处理版本冲突，生成 $(2, 2, 1)$ ；在f时刻，节点3上的数据再次被更新，版本信息变为 $(2, 2, 2)$ ；接着该版本传播到节点1上，覆盖掉原来的 $(2, 0, 0)$ ；在g时刻，数据版本更新为 $(3, 2, 2)$ 。通过这样的方式进行同步和更新，达到节点之间的数据一致性。

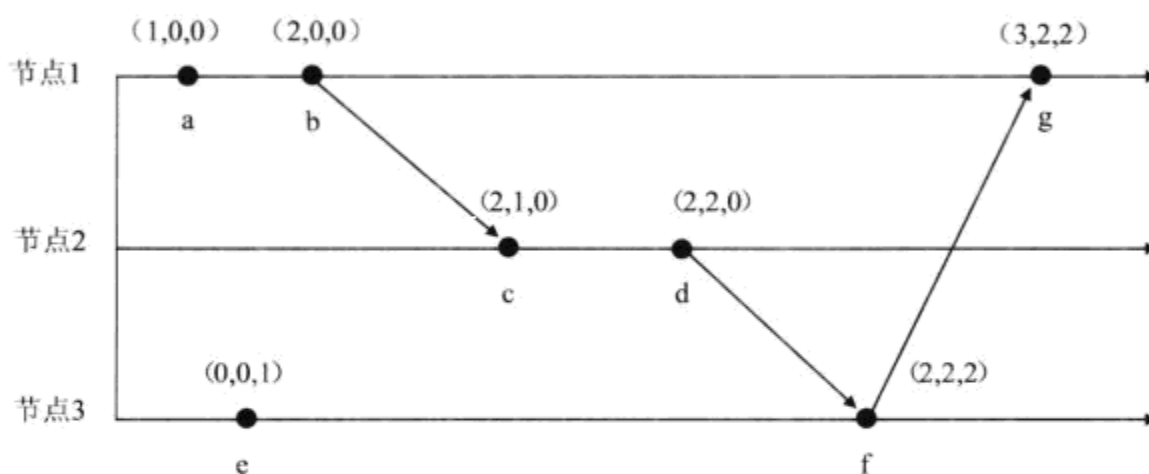


图9.5 数据一致性示例

9.3.1.3 非关系型数据库的数据节点管理

目前，大多数非关系型数据库是由多个地位平等的节点组成的，这些节点通过网络连接在一起。为了保证整个系统健康运行及存储数据的一致性，节点管理是必不可少的。下面介绍节点管理的一些主要方面：消息通信、失效检测及处理、节点添加与删除、持久化存储。

为了实现节点管理，节点间的消息通信是必不可少的。在分布式的环境下，如何能够有效地实现通信呢？解决这个问题的基本方法仍然是一个经典方法——Gossip协议，不过不同的实现对它进行了改进。在这里只是介绍一下它的基本概念。Gossip协议的思想源自于自然界中的病毒传播模式，在分布式环境下，每个节点都维护了状态信息及相应的版本，这些状态信息包括了用于判断节点是否存活的心跳信息、用于记录节点负载的信息，以及分区信息等。所有节点都会每隔一段时间随机选取一个节点进行通信，比较两个节点的状态信息，从而更新过时信息。通过这样的方法，在节点数为 N 的系统中，达到同步所需要的时间在 $\log(N)$ 数量级，满足了弱一致性前提下的要求。

有了Gossip协议作为分布式环境下节点通信的基础，失效检测就能很好地解决了。不过，判定失效也有多种实现。在Dynamo的实现中，如果某个节点（假设为节点3）在一定

时间内没有响应同步信息，发起同步信息的节点（假设为节点1）便认为该节点失效。在Cassandra中，判定节点失效采用了另外一种方法——基于疑似度的检测（Accrual Failure Detection），其基本思想是在分布式环境下，由于网络丢包或负载原因，判断一个节点是否失效不能简单地裁定为是或否，而是通过一个值（也就是疑似度）来判断。在一定的时间窗口内，如果节点未能响应同步消息一次，那么它的疑似度就增加一分，相反，误判几率就小一些，达到某个阈值后，系统就确定它失效了。

当然，节点失效有可能是暂时的，也可能是永久的，这两种失效的处理方法有所不同。在节点3失效的这段时间内，节点1会定期尝试连接它，并且所有发往节点3的数据更新操作都会暂时转发到其他的节点上保存。如果和节点3的通信恢复，那么更新的数据将同步到节点3上，从而避免了节点或网络的暂时失效对整个系统的可用性造成的影响，这个技术也被称为“提示移交”（Hinted Handoff）。如果确定节点3永久失效了，则需要人工将其从分区中删除，并且重新分配曾经由节点3负责存储的数据，相关节点还需进行数据同步。

为了实现快速地实现数据不一致检测，非关系型数据库常常采用Merkle Tree算法。简单来说，每个节点都维护了一个Merkle Tree，这颗树的叶子节点是它所负责的分区的数据的哈希值，树上其他节点都是其子节点的哈希值再取哈希值。在比较数据是否一致时，只需要查看两个负责相同分区的节点比较它们Merkle Tree根节点的值是否一致，如果不一致，再查找子节点，如果Merkle Tree的节点数为N，那么最多采用 $\log(N)$ 的时间就能够找到数据不一致的分区。

如图9.6所示，节点1和节点2具有相同的分区范围，通过比较它们Merkle Tree的根节点，发现哈希值不同，然后最终发现分区3的数据在两个节点上不一致。

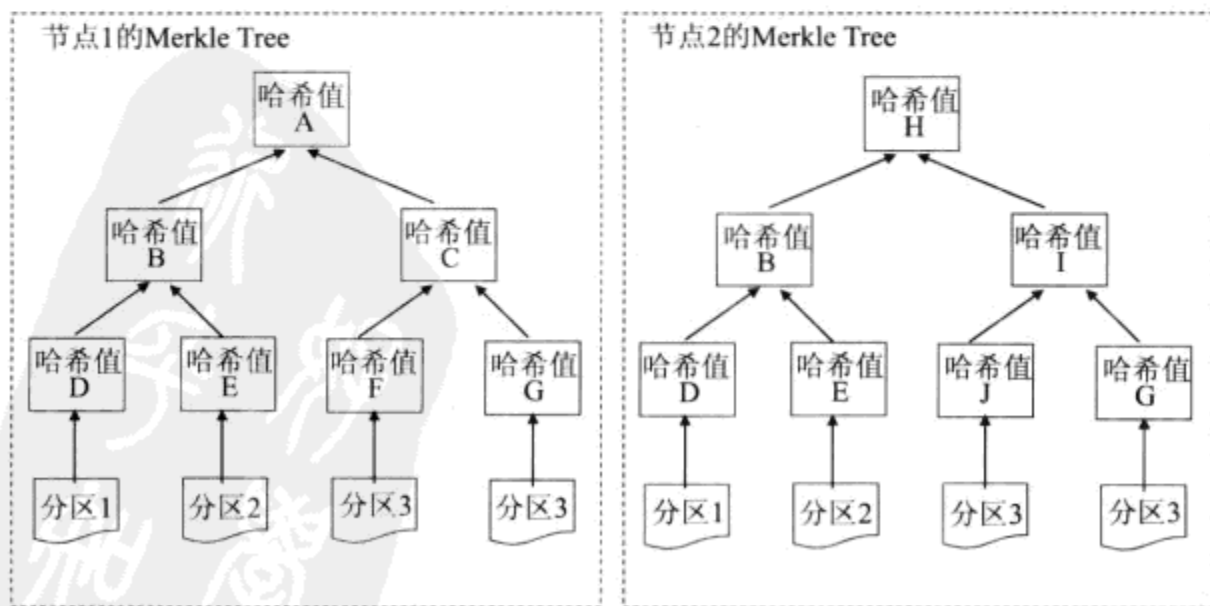


图9.6 通过Merkle Tree来比较数据一致性

当系统负载过重时，需要添加新的节点来分担负载，当某个节点发生不可恢复的错误时，需要将该节点从系统中移除，这些都会引起整个系统的分区发生相应的变化。一个新节点的加入需要完成如下几步：

- (1) 启动前的配置；
- (2) 启动自举；
- (3) 提供服务。

在第一步中，节点需要选择对应的哈希值，既可以是人为指定节点，也可以是通过哈希函数产生的，另外，为了和其他节点进行通信，它还需要知道系统中部分节点的信息，我们将这些节点称为种子（Seed），这些主要配置工作完成以后就可以启动节点了。在第（2）步自举过程中，新节点将自己的位置（哈希值）及分区信息告诉种子节点，种子节点发现这是一个新的节点，然后这样的信息在节点间通过Gossip协议传播，经过一段时间，系统中的所有节点都认识了该新节点。另外，由于它的加入，分区发生了变化，以前由其他节点负责存储的范围划归为新节点负责，因此数据将从这些节点转移到新节点上，在该过程中新节点不会对客户端负责数据更新，直到数据转移完成。自举完成以后，所有节点的信息都更新了，新节点也可以提供服务了。

为了可靠地存储数据，节点需要将数据存储持久化层中。每个非关系型存储系统采用的方法也不尽相同，比如BigTable就利用了Google File System，Dynamo利用了Berkeley DB，而Cassandra就依赖于本地的文件系统。另外，几乎所有系统都通过提交日志（commit log）的方式在持久层记录对数据的操作，如果节点进程意外崩溃，新起的进程可以根据这些日志对数据进行重新操作，从而很快恢复正常工作。

9.3.2 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9.7所示。

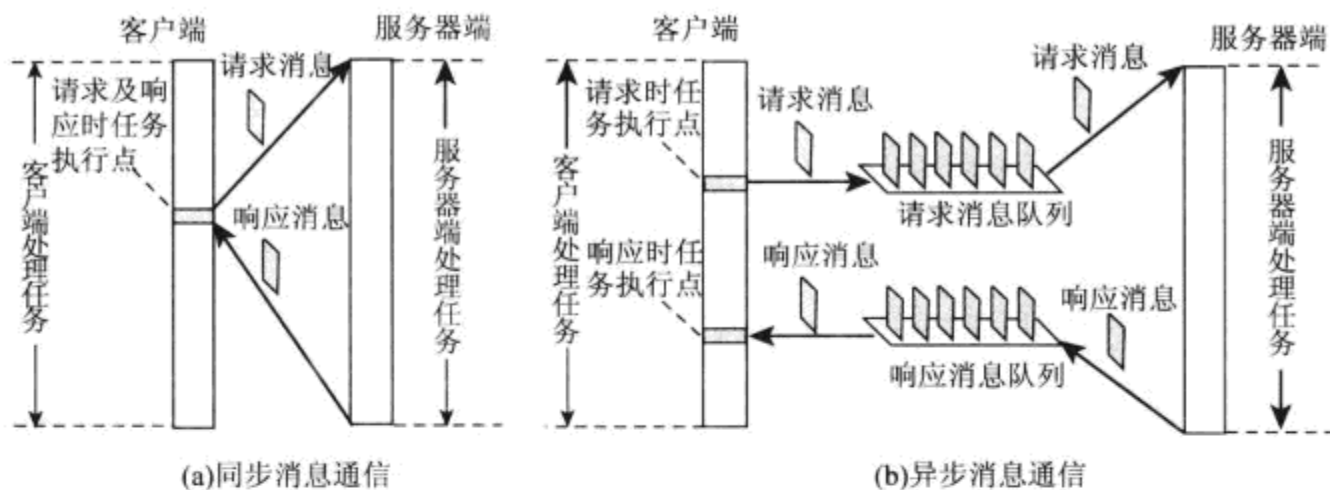


图9.7 同步消息通信和异步消息通信

在同步消息通信中，客户端任务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；在服务器端，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器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并发处理任务；

其次，同步通信机制造成服务器端的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，客户端和服务器端并不直接通信。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年就有人提出了基于生产者/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。一旦消息传送出现问题，需要有技术能够保存消息，并不断重试传送，等待故障被修复后再次进行通信，这样就需要有消息的保存机制、冗余备份机制、副本同步机制等。

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，为此必须要支持多种方式的消息读写模式，如边读边写、边读边发送、边写边发送等。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。云中的消息通信还要能够支持各种各样的消息格式，兼容多样的消息长度，因为使用云的用户并不局限于某一领域或者某一平台。

9.3.3 海量数据分析

以互联网为计算平台的云计算，将会更广泛地涉及海量数据处理任务。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。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：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key/value（键/值）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9.8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。

（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子任务完成后，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。MapReduce工作原理如图9.8所示。

当然，MapReduce的原语设计也会导致性能问题。美国加州大学伯克利分校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。因此该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 $M \times N$ 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技术的系统需要考虑的技术挑战。

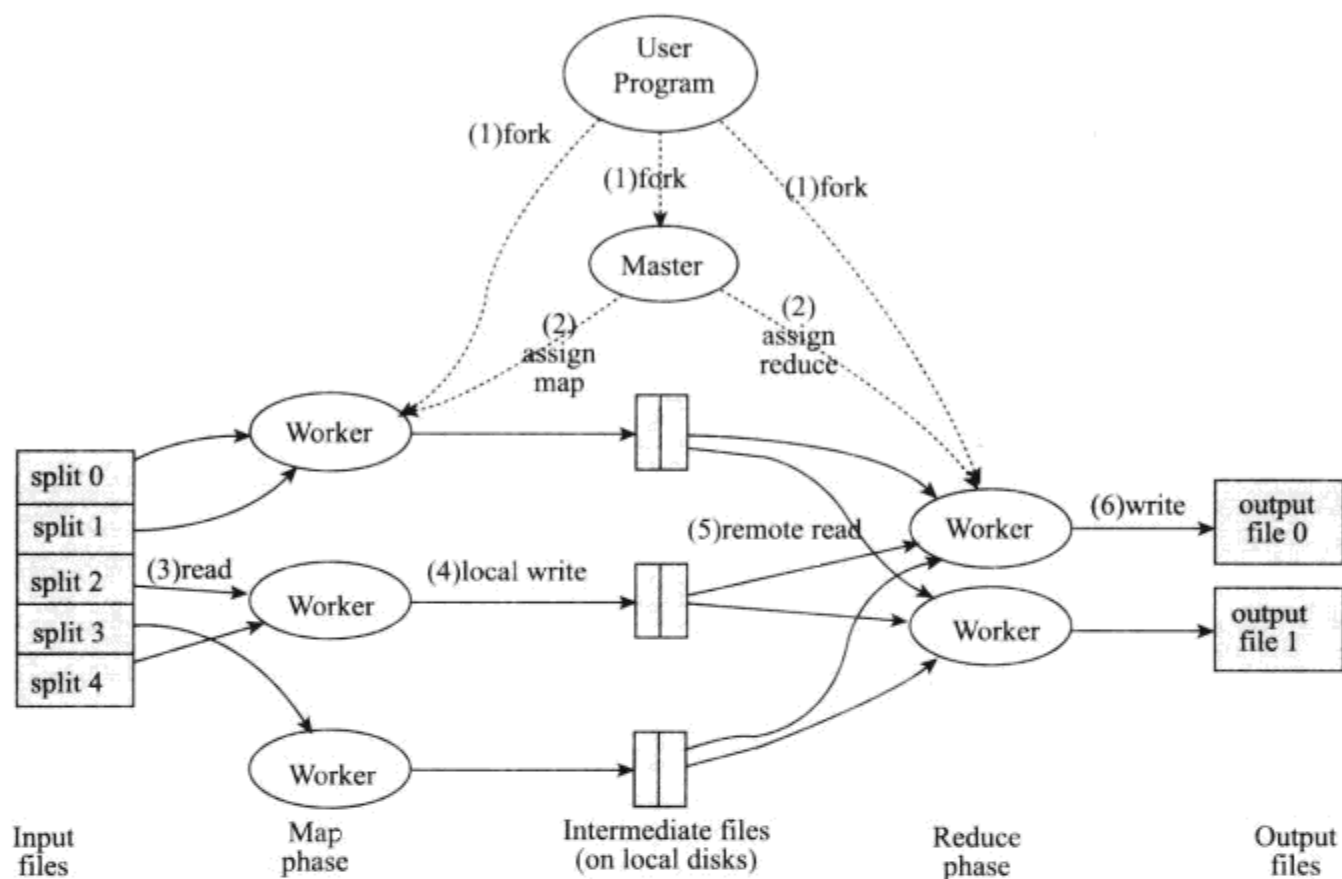


图9.8 MapReduce工作原理

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。解决这个问题的方法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位置感知的网络存储系统，这样就可以缓解上述性能问题。

大规模数据处理的另一类新技术被称为流计算（Stream Computing）。传统的计算或者数据处理的步骤是：首先收集数据，然后将数据储存起来，储存方式可能有数据库、文件等，然后对储存好的数据进行计算处理，将计算结果返回或者输出。这种计算模式，在现在的大量数据情况下遇到了挑战：

(1) 数据量非常大, 目前互联网时代的数据不止是文本数据, 而绝大多数是图像、音频、视频等, 将这些数据进行存储就是很大的问题, 针对海量数据进行计算也是很大的问题;

(2) 用户对于计算速度的要求越来越高, 比如天气预报、金融分析、市场预测等等应用, 要求数据计算产生结果的速度尽量快;

(3) 大量数据是实时生成的, 比如用户的使用日志、交通流量实时监测、证券实时报价等, 对这些数据进行分析计算的结果, 很多时候都需要实时可用, 而如果等待存储—计算—输出的过程, 则无法满足需求。

流计算是正式应对这些挑战的一项新兴技术。流计算的计算模式为: 数据实时地进行输入, 不需要强调存储过程, 实时计算, 实时输出。流计算将计算过程转化成一个流程图的形式, 每一个计算模块负责流程中的一个步骤, 通过网络连接将这些模块串接成一个反映整个计算过程的图, 图的起始端就是用户输入的数据, 而图的终止端就是计算输出的结果。流计算着重在“实时”上, 以此来解决上文提到的传统计算和数据处理面临的挑战。

目前, 流计算还在研发阶段, 需要研究对不同类型数据的处理方式、处理性能、硬件架构、软件支持等问题, 同时也需要业界逐渐接受这种新兴的计算模式。目前, IBM、Google、Nvidia等公司都在进行流计算的研发, 并推出了一些产品, 例如IBM的System S。

9.4 参考实现

当前, 运行在PaaS上的应用主要有两类: 一类是Web服务, 一类是数据分析服务。前者指通过浏览器访问的采用请求/响应模式进行交互的应用。这类应用长时间运行着, 要求保障响应时间、高可用性等。后者指对大量数据进行的分析处理。这类应用需要大型的计算能力和存储能力, 对于实时性的要求不高, 数据处理完毕后就结束运行。

市场上针对这两类应用的PaaS服务也有很多, 如针对Web应用有Google App Engine (GAE), Microsoft Azure; 针对数据分析主要是基于MapReduce计算模式的数据分析平台, 如Hadoop平台和Amazon的基于Hadoop的Elastic MapReduce服务等。本节将分别介绍面向这两类应用的PaaS平台结构和功能, 以让读者对于PaaS的机制有更深入的理解。

9.4.1 事务处理类

在当前的云计算实践中，基于互联网所提供的公共服务平台，比如GAE等，已经得到了相当的认可和应用。但是在企业环境中，普遍的做法还是局限于通过采用IaaS相关的技术对基础的计算资源进行虚拟，而很少将企业的应用运行在PaaS上。究其原因，主要还是在类似GAE这样的系统由于采用了新的编程模型和接口，现存的企业应用无法直接运行在这类PaaS上，需要重新进行投入进行软件的开发和迁移。除去这些额外开销所带来的经济因素之外，对这些已经稳定运行的应用进行重新开发也会带来额外的风险。这里提供一个面向企业的PaaS实现，这个实现实际上是IBM的企业级PaaS产品的一个原型版本。除了向大家展示一个事务型PaaS的基础组成部分之外，其设计目标主要有两点：首先是通过虚拟化和共享简化企业管理，降低企业的运营成本；其次是提供一个对现有企业应用兼容的PaaS平台，使当前存在的企业应用能够直接迁移到这个平台上。

9.4.1.1 系统概述

作为一个支持传统Java EE应用的平台，一般来讲，其架构中需要包含如图9.9所示的几个部分。其中每个部分的主要功能简单描述如下：



图9.9 事务处理类PaaS系统架构

1. 应用管理子系统

此子系统提供了对运行在平台上的应用的管理工具和服务。在用户使用PaaS的过程中，平台需要提供相应的工具对应用进行部署、管理和监控。由于运行在PaaS上的应用程序的多样性，一般来讲，PaaS会制定相应的UI框架和规范，根据用户的需求和应用所提供的一系列原数据来自动生成相应的用户界面。另外，除了针对每个应用的管理之外，系统管理员也需要对PaaS本身进行管理，比如配置应用的缺省策略和中间件模版，监视平台中各项资源的总体使用情况等。

2. 平台存储仓库

PaaS中需要一个全局的数据仓库用来存储平台用于管理自身及支撑应用的各种数据。比如，平台本身的核心程序，运行应用所需要的元数据和程序文件，以及各种中间件程序和原始的虚拟机映像等。除此之外，平台存储仓库也可以用于应用运行时的各项系统信息的存储，比如应用当前所占用的资源信息、管理所需要的系统日志等。

3. IaaS适配层

PaaS一般建立在IaaS之上，按需从底层的IaaS申请原始的计算资源。现在市场上存在不同的IaaS产品，虽然它们所提供的资源类型基本一致，但是其提供资源的方式和接口有很大区别。企业一般会根据自己的需要独立（于PaaS）地选择IaaS产品，为了最大程度地适应企业环境，企业级PaaS一般需要通过额外的IaaS适应层来为平台及其中的应用屏蔽掉IaaS环境的细节。

4. 核心服务

这里包含了PaaS运行所需要的所有关键功能的集合。比如为应用进行分配对应的计算资源，构造虚拟机上的软件栈等。PaaS的其他模块通过访问核心服务来实现相应的管理功能。

5. 应用程序运行时

包含了应用在运行时的所有相关的软硬件资源，即虚拟机、包含了中间件和PaaS管理应用所需要的一系列程序的软件栈等。与传统的概念不同，PaaS中的应用程序运行时是根据用户的需求和应用运行时状态动态改变的分布式系统，包含了负载均衡、防火墙等用户不可见的服务程序。

9.4.1.2 关键概念

与基于传统的中间件的企业应用模式不同，PaaS中应用的部署和管理的方式和概念都有了很大变化。这里列出其中一些主要的部分。

1. 应用模型

企业应用的需求一般分为功能性的需求和非功能性需求，比如可扩展性、响应时间等。在传统的中间件环境中，开发团队一般专注于程序的功能性需求进行开发。一般存在

一个专门的运营团队根据应用的非功能性需求，选择相应的硬件及各种软件产品，确定软硬件具体的配置等。在应用上线之前，运营团队制定相应的部署和后期的管理计划。随着企业应用规模的扩大，运营上的开销已经成为企业IT支出的一大部分。在以往的实践中，我们可以发现，大部分满足非功能性需求的解决方案实质上都是大同小异的，比如，通过负载均衡器来实现自动的扩展性，通过主从方式的配置来解决可靠性问题等。在PaaS中，通过这些解决方案总结为对应的模式，交由平台来自动实现，可以降低对运营团队的需求。

将非功能性需求提取出来使得开发和部署应用的复杂度和所需要的专业技能大大减少。用户只需要根据应用的功能性需求开发完成，然后通过PaaS所提供的建模工具以策略的形式指定应用的非功能性需求。PaaS在部署应用的时候，会自动实现支撑这些需求所需要的物理结构（即虚拟机及其中运行的软件栈）。同时，在应用运行的时候，PaaS会根据应用的运行时状态，动态调整这些支撑结构，以满足这些需求。

2. 软件栈

不同的应用所使用的中间件类型往往有一定差别，即使某些应用用到相同的中间件程序，其中的配置也往往差别很大。在传统的环境中，构造支撑不同应用所需要的软件栈往往是一项非常繁重的劳动。在应用需要扩展的时候，管理员很难向应用中迅速提供带有解决问题的软件栈的节点。PaaS需要能够自动根据应用的需求，动态获得计算资源（即虚拟机），在其中安装应用所需要的一系列软件程序。

在我们的示例系统中，中间件按照其功能，以插件的形式在虚拟机上进行组织。中间件程序作为插件，以模版的方式存储在平台存储仓库中。PaaS中存在一个关于软件栈的插件框架，根据每个虚拟机的任务按需从仓库中下载模版，然后实例化插入到框架中，以形成本机的中间件环境。这样的方式有如下几个好处，首先，不同应用之间，同一应用的不同虚拟机之间，都需要不同的中间件环境，插件的方式简化了中间件程序的安装；另外，应用往往不需要一个完整的中间件服务器。通过将中间件按照插件进行组织，能够进一步按照应用的要求提高资源的使用率。

3. 资源组织

前面已经提到过，PaaS实际上可以看成是应用相关的软硬件资源及其他服务的集成平台。这些资源和服务来自不同的层次甚至其他的服务提供商。PaaS需要提供给用户一个一致的、统一的视图以完成对这些资源和服务的监控。换句话说，PaaS需要提供给用户针对

应用及其部署的一个逻辑模型来对应用进行管理。在这个模型中包含了每个应用所用到的资源，屏蔽了诸如应用的拓扑、虚拟机数量等动态物理信息。

在我们的示例系统中，用一种面向资源的方式来实现这个目的。应用中所用到的各种（需要管理的）资源首先被分类，每类资源都通过插件的方式定义和实现了这类资源所支持的一系列属性、监控指标及操作。PaaS为每个应用自动生成所用到的所有资源的一个统一的界面，用户通过这个界面来对这些资源进行管理，每个管理操作会自动地被PaaS转发到相关的资源实例上。用户不需要考虑资源的定位和相应管理功能的调用等问题。

9.4.1.3 实现细节

前面两节介绍了本示例系统的基本结构和概念，本节进一步讨论本系统的一些实现细节，以展示PaaS的一些内部工作机制。

1. 应用实例化和激活

在用户完成应用的开发之后，需要本示例系统所提供的工具对应用的部署进行建模，即指定应用的功能性模块和相关的非功能性策略。这实际上是提供了针对应用的、与底层物理实现无关的逻辑视图。在应用能够部署到物理机或虚拟机上运行之前，需要针对这个逻辑视图建立起相应的物理视图。物理视图包含为应用分配多少虚拟机，每个虚拟机需要安装哪些软件等内容。这就是应用实例化的过程。

在实例化完成之后，应用还需要有一个激活过程，即修改软件的配置以适应应用运行时的动态信息。比如：在物理视图中，无法提前获知虚拟机通过动态地址分配所获得的IP地址信息，因此，在物理视图中，只能包含应用服务器和数据库服务器之间的逻辑连接信息，在应用能够正常工作之前，需要对相关服务器上的软件配置进行更改，使应用服务器能够获知数据库服务器的IP地址以正确地进行访问。

2. 代理框架

前面提到，本示例系统通过IaaS适配接口来获取原始的计算能力，即虚拟机。在获取虚拟机之后，这个时候的虚拟机，往往只有一个最基础的操作系统，需要根据应用对这个虚拟机的要求，从平台存储仓库中下载所需要的各类软件程序。另外，在应用运行的过程中，用户及平台本身都会对应用进行监控，发出指令到虚拟机，调用对应软件（或资源）的管理功能。这些功能在本示例系统中由一个代理框架来负责。

本示例系统中，每个虚拟机都会运行有一个平台所提供的代理程序。这个代理程序除了定期向平台的管理系统报告自身的状态之外，也会接收平台所发来的控制指令，比如：下载某个资源安装到本地，或者重新启动某个中间件程序等。一个应用的所有（虚拟机上的）代理会构建一个对等网络来进行代理之间的通信，在本地维护一个关于应用的统一的物理视图。另外，每个应用的代理会通过分布式的选举算法，选出一个（应用范围内的）主代理，作为应用系统控制消息的入口和分布式操作的协调者。

3. 资源配置和调用

PaaS一个重要特性就是使用户能够通过应用的逻辑视图进行管理，而不需要关心应用运行过程当中物理结构的变化，比如由于自动伸缩和可靠性策略所导致的虚拟机的创建和删除等。比如：用户有时候需要提高应用服务的性能，在PaaS中，他只需要在管理界面上更改应用逻辑视图中的应用服务器的性能级别，PaaS将会自动地将这个级别转换为中间件的新的属性（比如缓存和堆大小等）配置到所有的应用服务器实例之上，即使系统今后为这个应用分配新的应用服务器实例，也将直接使用更新之后的属性对新的实例进行配置。

本示例系统中所有插件都会有相应的元数据来指明每个逻辑模型中的对象及中间件程序都支持哪些属性和行为。PaaS会自动生成相应的界面提供给用户。同时，PaaS还会自动地将逻辑属性和行为转换为相应的中间件程序的配置和行为，通过PaaS所提供的分布式执行引擎在每个中间件实例上进行配置和执行。同时，PaaS也会在应用程序范围内保证应用的逻辑模型的一致性，即在新创建的中间件实例上应用最新的配置。

综上所述，我们叙述了本示例系统的重要设计原则和实现。简单来说，本示例系统中通过建立应用部署的逻辑模型，将应用的非功能性需求通过策略来表示。平台自动将这个逻辑模型转化为支撑应用运行的物理结构（即所需要的中间件程序及其配置）。另外，系统中的中间件系统以插件来组织，通过系统的代理框架根据应用的不同需求自动构造相应的软件栈。这些设计都使得以往需要大量人力和专业技能的运营工作交由平台来自动实现，大大简化了企业的IT运营，从而节省了企业在IT上的投入。

9.4.2 数据分析类

随着数据广泛可得、数据量迅速增加、从数据中获得知识的重要性不断显现，数

据分析的重要性也快速增加。当获取数据的广度和深度的扩展导致数据分析任务变得越来越多时，对数据分析的需求就从此前特定领域如科学研究和工程模拟等扩大到通用领域。例如，企业可以分析自身运行中获得的数据以了解企业运行状况，商场可以分析顾客的购买行为以优化商品配置，大型网站可以分析用户的访问行为以更好地了解用户的习惯和偏好。

当数据分析需求扩大之后，提供数据分析功能的系统就要考虑到分析需求的多样性、提交任务的随时性、数据格式的灵活性、分析方法的可变性等。更为重要的是由于数据分析涉及大规模的数据存储能力和计算能力，普通用户保有这些的存储和计算系统却并不时时使用，这是一种成本上的浪费。因此，从大众化的数据分析中抽象出通用的数据分析需求，设计能够适应大众需求的数据分析系统并通过服务的方式提供给用户使用，是数据分析类PaaS平台发展的根本动力。

9.4.2.1 系统概述

作为数据分析系统，在处理大众的数据分析任务时，需要具有的基本能力包括：

- ▶ 大规模的数据存储；
- ▶ 处理分析任务的大型计算能力；
- ▶ 容纳用户的多样性分析需求；
- ▶ 快速的资源调度和向多个用户同时提供服务的能力。

有了这样的数据分析系统，用户可以通过简单的接口向分析系统提交待分析数据及自己的数据分析方法，而数据分析系统调度计算资源执行分析任务并返回分析结果。由于数据分析涉及大规模的数据量和计算量，数据分析系统本质上就是一个大规模分布式数据存储和处理平台。

数据分析任务的特点不同导致数据分析系统的设计也不同。当前一系列的公开数据分析系统所针对的数据分析任务特点包括：

- ▶ 大规模的数据量，数据往往需要存储在多个节点上；
- ▶ 数据可以分块且分析任务可在各数据分块上并行执行；

► 在同一份数据上存在多种分析任务并多次执行。

这类数据分析任务已经总结为MapReduce模型。研究人员早在二三十年前就已认识到MapReduce模型，而Google将其应用到搜索引擎的大规模数据分析中并显示其威力。由于MapReduce模型以及其一系列改进模型能够适应一大类实际的数据分析需求，因此它得到重视并有多个平台支持这类分析任务。典型的平台包括Google所使用的数据分析平台和Hadoop。事实上，Hadoop是Google数据分析系统的开源实现。由于Hadoop的开源性，人们能够很容易地接触到它的功能和结构，因而得到相关人员的学习和使用。

Hadoop实质上是可靠的分布式数据存储和执行MapReduce类型分析任务的平台。从功能层次来看，Hadoop分为三个层次（如图9.10所示），最下面的是Hadoop Core，实现分布式系统的构建和维护。它实际上包括一组部件和接口实现分布式的文件系统和通用的I/O操作（如序列化、远程调用和数据持久化等）。在Hadoop Core之上提供三个方面的基本功能，分别为实现数据存储的Hadoop分布式文件系统HDFS，实现数据分析的MapReduce计算框架，以及实现分布式系统构建和管理所需通用功能的Zookeeper。在这些基本功能之上，Hadoop提供一系列的工具，如PIG、Chukwa、Hive、HBase等。这些工具开发和使用基于Hadoop的数据分析更为方便，用户也可以进一步定制从而实现自身所需要的功能。

从用户角度讲，Hadoop暴露的核心功能是HDFS和MapReduce，而Zookeeper在需要自己构建基于Hadoop的分布式应用时也会涉及。下面分别就HDFS、MapReduce和Zookeeper逐一做简单介绍。

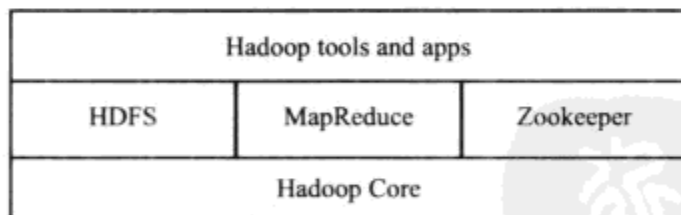


图9.10 Hadoop系统的功能层次

9.4.2.2 HDFS模块

HDFS是Hadoop的分布式文件系统，实现大规模数据可靠的分布式读写。HDFS针对的

使用场景是数据读写具有“一次写，多次读”的特征，而数据“写”操作是顺序写，也就是在文件创建时的写入或者在现有文件之后的添加操作。HDFS保证一个文件在一个时刻只被一个调用者执行写操作，而可以被多个调用者执行读操作。

文件系统通常采用“块（Block）”的概念作为数据读写的操作单位。一个磁盘上的“块”通常为512Byte，而一个本地文件系统上的“块”通常为若干KB。HDFS也采用“块”作为其管理的文件存储单元。HDFS的“块”通常为64MB或者128MB。HDFS中的一块数据位于一个节点上，而一个大型文件可能存储在多个块的单元中，一个文件所在块可以位于不同的节点。HDFS系统包括三类节点：HDFS客户端、命名节点（Namenode）和数据节点（Datanode）。客户端是用户与文件系统打交道的接口，它提供读写文件、管理文件的接口操作。命名节点用于维护文件系统的结构和元数据，例如文件系统的目录层次、文件信息、数据存储位置等。而数据节点是实际的数据存储地，以块为单元保存数据。

图9.11和图9.12分别展示了在HDFS中进行数据读取和写入操作的基本流程。HDFS客户端通过与命名节点的交互获取关于目录和文件的元信息，通过这些元信息就知道实际数据的存储位置，然后与数据节点进行交互执行文件数据的读写操作。为了保证数据的可靠性，HDFS可以设置数据的备份数，即同一份数据保存在多个数据节点，从而容纳节点的失效。

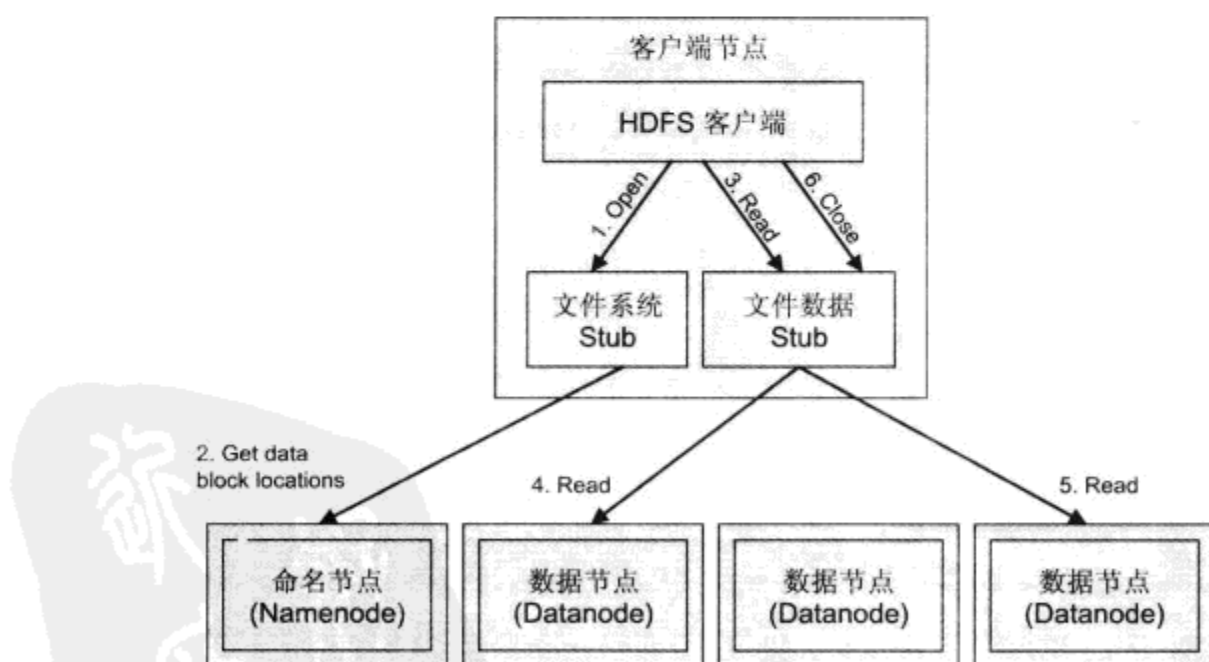


图9.11 HDFS读数据的流程

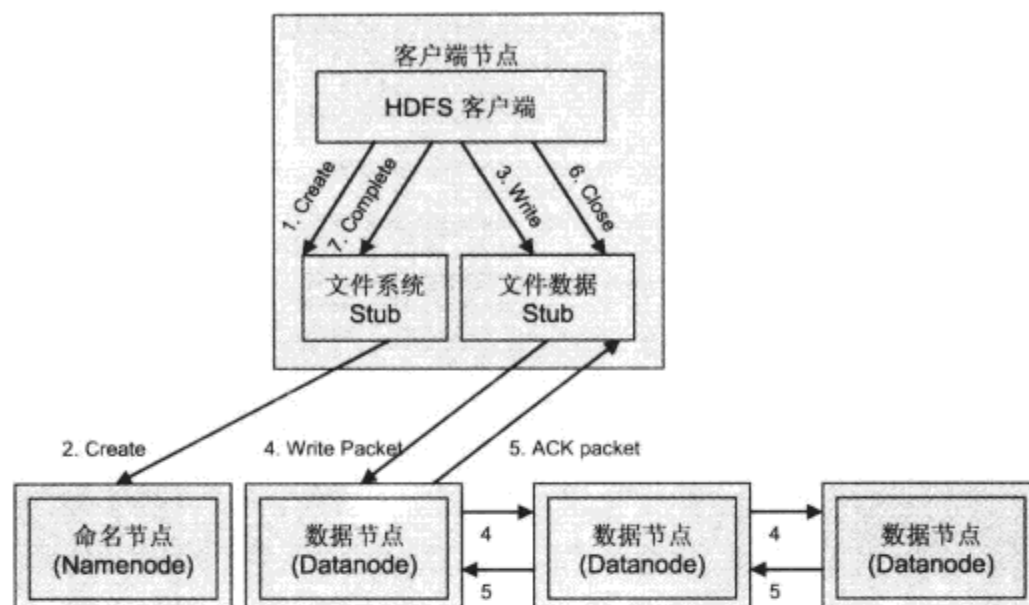


图9.12 HDFS写数据流程

由于数据保存在多个位置，维护数据的一致性就是HDFS系统的一项重要任务。因为HDFS支持的是顺序写入方式，它支持简单的一致性操作。当一个文件创建并保存后，它对“读”是可见的。但正在向一个文件写入时，当前被写入的“块”对“读”是不可见的，其他已经写满的块对“读”是可见的，除非执行了同步（sync）操作。也就是说，在写入数据时，如果调用了“sync”操作且“sync”操作返回成功，到此刻写入的所有数据都会持久化到存储系统中且对所有的“读”可见，也包括当前正在操作的数据块。因此，在HDFS中，数据读取可能落后于数据写入，这一点值得编写分析程序时留意。

HDFS中的命名节点维护文件系统的逻辑结构、文件的元信息包括文件数据实际存储位置信息，它是文件系统可访问的关键。命名节点将文件系统的信息存储在内存中以供快速访问和操作，同时将数据持久化到本地文件系统和远程文件系统（如NFS）以备故障恢复。同时，HDFS也可以使用备份命名节点并与主命名节点实现数据同步，以便主命名节点失效后备份节点快速接管主命名节点的任务。选择性能好、可靠性高的服务器充当命名节点也可以提高HDFS的可用性。

HDFS还使用一系列优化措施来提高系统的可靠性、可用性和访问速度，读者可以参考关于HDFS的文档或者直接阅读HDFS的实现代码。

9.4.2.3 MapReduce模块

MapReduce既是一类数据分析任务的编程模型，也是Hadoop中实现这类数据分析任务的功能模块。关于MapReduce模型的介绍可以参看9.3.3节，此处仅介绍Hadoop如何执行MapReduce任务。

图9.13显示了Hadoop执行MapReduce的流程。如图9.13所示，一次数据分析称为一个MapReduce工作（Job），分为Map和Reduce两类操作，这些操作根据对所操作数据的划分而分成多项任务。为了执行MapReduce工作，Hadoop系统中有两类节点：jobtracker和tasktracker。Jobtracker用于接收所提交的MapReduce工作，将工作划分为相应的任务并指定所操作的数据，它同时调度任务的执行并派发到相应的tasktracker节点，然后监控这些任务的执行过程，如果所有的任务都执行成功或者某些任务执行失败达到返回的条件，则返回工作执行的结果。Tasktracker节点监控自身的负载，在有空闲处理能力时向Jobtracker请求新的任务，在得到任务时启动新的JVM线程执行对应的任务，在任务执行过程中定期向Jobtracker汇报任务执行进展，在任务执行结束后返回任务执行的结果。

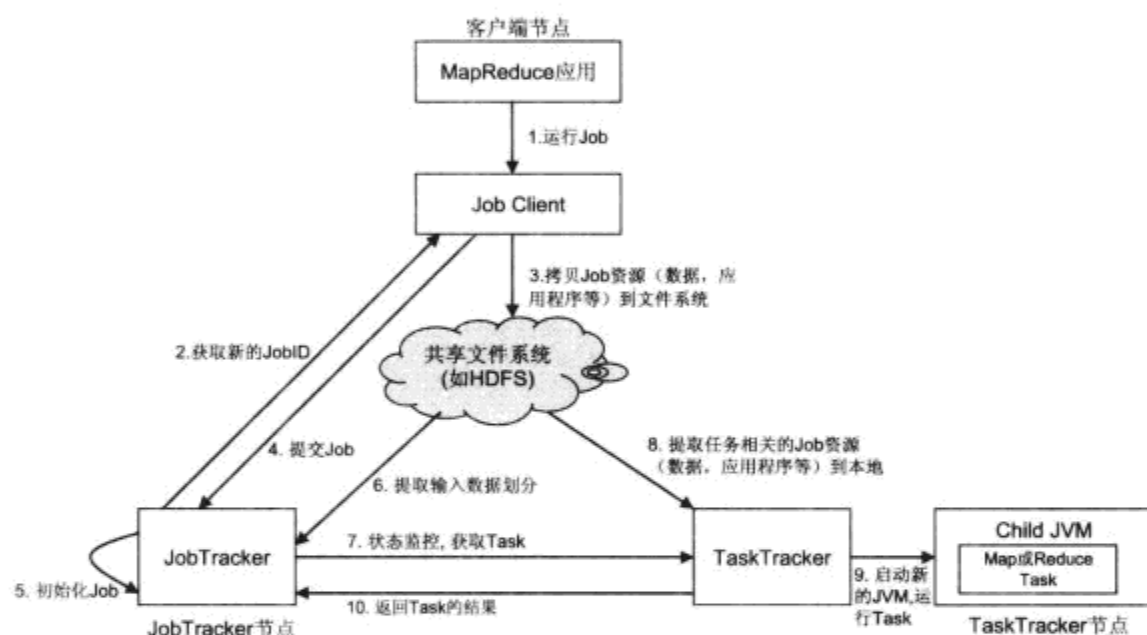


图9.13 Hadoop执行MapReduce工作的流程

为了保证工作执行的速度和可靠性，Hadoop系统有一系列的优化措施。例如，在划分Map任务时，每一个Map任务所操作的数据量尽量与HDFS所存储的数据块相一致，且把map任务尽量调度到存储了相应数据块的节点，从而使得数据读写发生在本地，减少了读取数据的时间和网络带宽消耗。用户可以配置jobtracker当一个任务在某一个节点执行失败之后，jobtracker将其调度到其他节点再次执行，从而减少整个工作返回失败的可能；如果一个任务执行速度过慢，jobtracker还可以在其他节点启动同样的任务，以期减少该任务的执行时间。每一个执行任务节点上的tasktracker启动新的JVM进程执行对应的任务，以减小由于任务实现代码中的错误导致该tasktracker崩溃的概率。Tasktracker还可以在一个JVM中执行多个来自于同一个工作的任务，从而减少启动任务执行的时间。Hadoop在MapReduce工作执行的过程中还采取了一些措施来处理中间数据，以减少中间数据在Map和Reduce任务之间的传输量并加快处理速度。总之，Hadoop要努力做到的是通过这些并行化操作来提

高数据分析的速度和可靠性。

9.4.2.4 Zookeeper模块

大型分布式应用需要基于一组基本服务用以维护分布式系统结构，以及协调任务执行和信息交换。在Hadoop系统中，有一个功能部件即Zookeeper来实现该类基本服务。Zookeeper类似于Google数据分析平台中的Chubby服务，是其开源实现。ZooKeeper以Zab算法为基础，Zab是一种全序广播（Totally ordered broadcast）协议，实现多个分布式节点之间的协同机制。Zookeeper可以提供分布式应用所需的同步（synchronization）、配置管理（configuration maintenance）、群组（groups）及名称服务（naming）等。

在Zookeeper中，数据以znode的形式组织成树形结构。znode类似于文件系统中的节点，该节点有名称属性及附加的数据。Zookeeper对一个znode的数据操作保持原子性，即整个读取或者完全替代一个节点的数据。Zookeeper还支持观察者（Watcher）操作，当所观察的znode发生改变时，系统会向观察者发送通知事件。

Zookeeper系统包括两部分：服务器和客户端。客户端连接到某一个服务器，通过心跳消息保持与服务器的连接，向服务器发送请求、接收服务器响应，获取服务器事件通知。客户端通过配置知道一组服务器，当与所对应服务器的连接断掉后，它可以转而连接到其他的服务器。Zookeeper的服务器之间通过选举机制选出领导，由领导节点来维护系统中的数据一致性。如图9.14所示，客户端所写入的数据通过领导节点发布到其他节点。Zookeeper采用的是多数一致原则，即领导节点将数据提交到多数节点上保存，则数据的写操作即成功。在数据读取时，Zookeeper的客户端节点直接从所连接的服务器读取，以分散数据读取的负载。

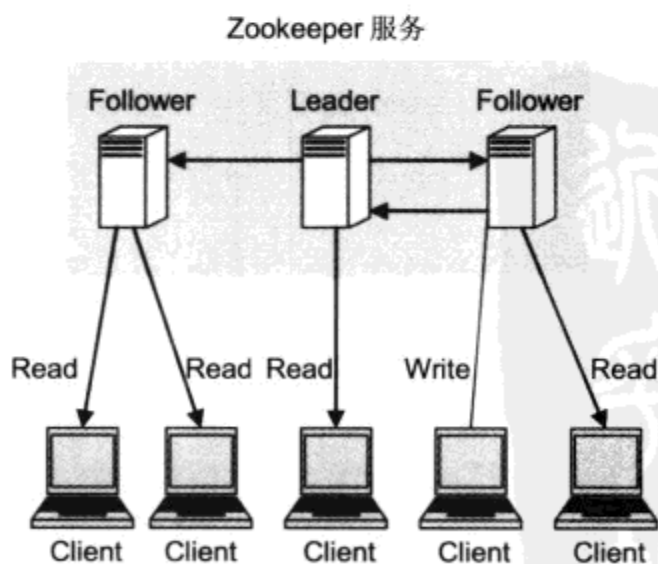


图9.14 Zookeeper中的基本结构

基于Zookeeper系统可以实现一系列的分布式数据结构和服務。例如，分布式应用的节点作为客户端把配置信息保存在Zookeeper中，它可以查看、修改甚至删除这些数据，而观察这些数据的其他客户端在数据发生变化时能够及时得到通知，由此实现了配置信息的管理和发布，并保证数据的可靠和及时性。

作为可靠的分布式数据结构和基础服务，Zookeeper已被Hadoop及相关的Hive、PIG等采用，它也可以单独用于构建其他的分布式应用。

9.4.2.5 公用Hadoop服务

用户可以选择自己搭建并使用Hadoop进行数据分析，也可以采用公开的Hadoop或者MapReduce服务来进行数据分析。Amazon于2009年推出基于Hadoop的MapReduce服务，称为Amazon Elastic MapReduce。Google App Engine的SDK中内置了对MapReduce的支持。在Microsoft Azure上也有Hadoopdotnet项目将Hadoop迁移到.NET平台上，而MySpace Qizmt是直接使用.NET实现的MapReduce框架，可以运行在Azure上。

一般来说，使用Hadoop（或者说MapReduce类数据分析服务），要考虑的因素包括数据存取、分析逻辑及分析任务调度。对于不同的平台，其采用的数据存取服务不同，而分析逻辑即Map函数和Reduce函数的需要根据平台所提供的开发接口来实现。分析任务的调度则由平台来负责，用户无须干预。因此，用户在使用基于MapReduce的数据分析服务时，应该检查所提供的服务提供的数据存取方式，使用编程接口构建Map和Reduce函数的便利性。

以Amazon Elastic MapReduce为例，我们可以了解使用这类服务的基本步骤。Elastic MapReduce服务通过Amazon S3服务来存取用户提交的数据和分析应用代码，通过Amazon EC2实例来执行数据分析的计算，而MapReduce服务平台则负责分析任务的划分和调度。用户可以通过以下几个简单的步骤来使用Elastic MapReduce服务：

- ▶ 开发所需的数据处理程序，主要是实现Map和Reduce函数；
- ▶ 上载数据和数据处理程序到Amazon S3中；
- ▶ 登录到AWS管理控制台启动Elastic MapReduce工作流，指明所需EC2实例类型和数量，指明处理程序和数据在Amazon S3中的位置；

- ▶ 从AWS管理控制台监控 workflow 执行过程，执行结束后从Amazon S3中获取所获得的数据分析结果。

采用公开的Hadoop服务，用户无须关注数据存储和计算所需的资源，无须考虑Hadoop平台的管理和维护，而可以将精力集中在数据分析逻辑本身。因此，如果用户的数据分析任务不是过于庞大、对数据隐私和安全要求不高，且需要尽快获得分析结果时，他们可以考虑尽量采纳公开的Hadoop服务。

9.5 小结

平台即服务将云计算平台层的功能以服务的形式提供给用户，用于应用的开发、发布和运行管理，是一类重要的云计算功能集合。本章首先阐述了平台即服务产生的驱动力，即其设计的基本需求，以及所支持的基本应用类型，帮助读者对PaaS的功能特征有清晰的了解。然后，本章介绍了实现PaaS系统的几项重要组成部分，它们是针对PaaS的功能和使用方式而专门设计的，根据其所处系统架构的位置，划分为核心系统和扩展系统。最后，本章针对平台层的两种主要运用形式，Web服务和数据分析服务，通过实例来剖析面向这两类服务的PaaS实现要点，帮助读者对平台即服务获得更深刻的理解。



第 10 章

软件即服务

- 10.1 概述
- 10.2 支撑平台
- 10.3 云应用
- 10.4 小结



作为一种软件交付模式，软件即服务（Software as a Service, SaaS）方兴未艾，成为一种不断扩展的软件使用模式。前述两章节已向大家介绍了基础设施即服务（IaaS）和平台即服务（PaaS），即硬件平台和服务管理平台。这就如同画家具备了光线充足的房间、平整洁白的画布和丰富的原料素材，而如何创作出令人赏心悦目的作品，仍需要他的创意和技术。这个创作过程，则是软件即服务层和SaaS应用开发者们需要面对的。

值得注意的是，不同于基础设施层和平台层，软件即服务层中提供给用户的是千变万化、层出不穷的应用。每一个成功的应用都有它的突出点，如满足用户需求的巧妙创意。软件即服务的基本功能就是要为用户提供尽可能丰富的创新应用，为企业和机构用户简化IT流程，为个人用户提高日常生活方方面面的效率。我们将这些应用称为各种各样的“艺”。但是，这些应用也一定有它们的共同点，那就是让应用能够在云端运行的技术。通过多年的实践，业界将这些技术或者功能总结、抽象，并定义为SaaS平台。开发者可以方便地使用SaaS平台提供的常用功能，减少应用开发的复杂度和时间，而专注于业务自身及其创新性。对于这个SaaS平台，我们将其称为“技”。

本章，仍然遵循云架构中自下而上的逻辑，首先讲述“技”，即SaaS平台，讨论它所应遵循的架构设计及涉及的关键技术，然后分享“艺”，即SaaS平台上运行的各种各样的云应用，讨论它们的特征和分类。在这两个方面，我们都会给出典型示例，方便大家感性地了解。

10.1 概述

软件即服务层是SaaS平台与云中应用的集合，这些应用以SaaS的模式交付。回顾第1章所介绍的概念，SaaS让用户无须再在自己的硬件上安装购买了许可证的软件，而是可以采用按需付费的模式，使用浏览器或者软/硬件终端，通过互联网访问应用。这种应用通常采用订阅的付费模式，而应用的维护、更新和技术支持则是应用提供商所提供的一部分。从应用提供商的角度来看，由于应用运行在自有或被托管的专业SaaS环境中，它能够更加有效地保护知识产权，降低测试和维护的复杂度，并通过多租户技术降低运营成本。

10.1.1 生态系统与实现层次

不难理解，如果SaaS应用在功能上存在着共性，那么就会出现细化的分工和层次。图10.1的左半部分展示了软件即服务的生态系统。可以看到，在既有的IaaS和PaaS平台提供商之上，会出现一个新的角色，SaaS平台提供商。它能够为SaaS应用提供通用的运行环境或系统部件，如多租户支持、认证和安全、定价与计费等功能，使运行在其上的SaaS软件提供商能够专注于客户所需业务的开发。

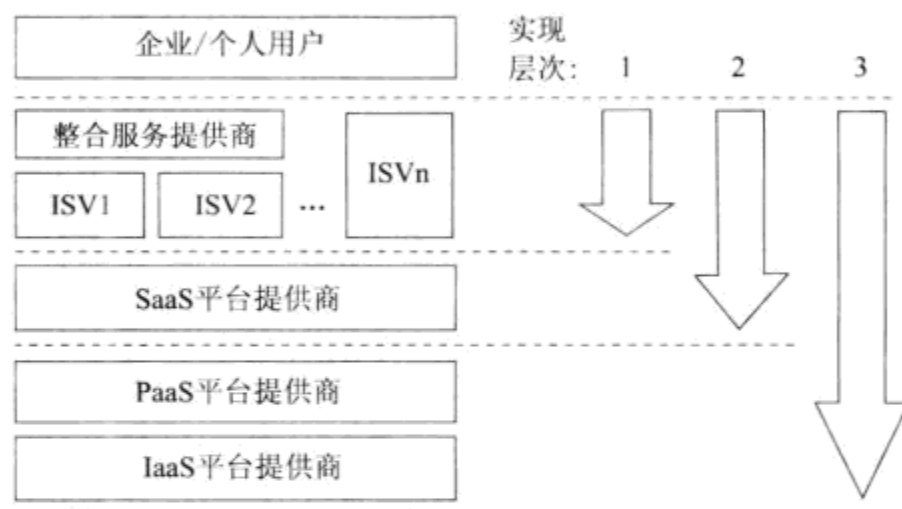


图10.1 SaaS生态系统和实现层次

运行在SaaS平台之上的是独立软件提供商（ISV），它们快速地实现客户的需求，以SaaS的模式交付软件功能。正如上文所述，整合是每个ISV都将面对的问题，也将出现专业的服务提供商将其解决。此外，与整合服务提供商相似，一类新的角色——应用迁移商也在SaaS生态系统中出现了。虽然没有在图中展示出来，它们能够帮助软件公司将现有的传统应用迁移到SaaS平台上，使软件能够顺应历史潮流，获得更强的生命力。SaaS的最终消费者来自于企业与个人用户。至此，一个完整的软件即服务层生态系统展现在我们面前。

回顾本书第1章所介绍的云计算的产业结构，图10.1将云服务提供商和应用提供商展开，细分了产业链条。在现实世界中，不同的提供商会选择自己不同的切入点，或仅针对其中一点来优化投资，或进行垂直整合来获得更大的控制力和更广的影响。以SaaS应用提供商为例，为了向用户交付一项软件功能，它们可以选择不同的实现层次，如图10.1右侧所示。

在第一类实现层次中，应用提供商依靠SaaS运行平台实现应用的交付，专注于用户需求。这种方式会牺牲一定的系统灵活性和性能，但是能够以较低的投入快速实现功能，适用于规模较小或正在起步的公司。在第二类实现层次中，应用提供商依靠云中的应用环境

和系统资源来搭建完整的SaaS应用。这种方式较第一种方案对应用提供商的要求更高，但是也赋予其更强的控制能力，使其能够针对应用的类型来优化SaaS基础功能，适用于规模较大、相对成熟的公司。在第三类实现层次中，应用提供商不依赖于云中的服务，在自有的资源和运行环境上提供SaaS应用。采用这种实现层次的公司往往具有雄厚的资金和技术实力，它们不仅可以为最终消费者提供服务，也可以作为运营商为在其他层次实现SaaS的公司提供平台服务。可见，在SaaS生态系统中，不同类型的公司都可以找到适合自己的实现层次，为用户提供不同类型的应用。

10.1.2 技术发展历程

“软件即服务”这个理念本身并不是新兴产物。在2000年左右，SaaS作为一种能够帮助用户降低成本、快速获得价值的软件交付模式而被提出。在近十年的发展中，SaaS的理念不断丰富，应用面不断扩展。随着云计算的兴起，SaaS作为一种最契合云端软件的交付模式成为瞩目的焦点。在由Saugatuck技术公司撰写的分析报告“Three Waves of Change: SaaS Beyond the Tipping Point”中，SaaS的发展被分为连续而有所重叠的三个阶段：

第一个阶段为2001年至2006年，被称为“有成本效益的软件交付”。在这个阶段，SaaS针对的问题范围主要停留在如何降低软件使用者消耗在软件部署、维护和使用上的成本。据IDC统计，在商业环境中，IT预算的80%被消耗在硬件、人员和技术支持上，只有20%被真正用于购买软件所提供的功能上。这个阶段的SaaS应用在一定程度上解决了以上问题，但仍有其局限性，比如，虽然具有了多租户的能力，但是仍然是一个封闭而孤立的应用，无法与企业现有的系统和业务整合。

第二个阶段为2005年至2010年，被称为“整合的业务解决方案”。在这个阶段，SaaS理念被更加广泛地接受，并且开始在企业的IT系统中扮演越来越重要的角色。如何将SaaS应用与企业既有的业务流程和业务数据进行整合成为这个阶段的主题。在这个阶段，SaaS应用越来越成熟，并且开始进入主流商业应用领域；SaaS应用的生态系统逐渐形成，提供应用整合的平台厂商开始出现。

第三个阶段为2008年至2013年，被称为“工作流使能的业务转型”。在这个阶段，SaaS应用的生态系统逐渐成熟和完善，成为企业整体IT战略的关键部分。SaaS应用与本企业的传统应用已完成整合，企业间的数据与业务的整合成为主流。SaaS从“被整合”的角色转型为“整合者”的角色，使企业的既有业务流程更加有效运转，并使新创的业务成为

可能。

云计算的出现为当前正处于由第二阶段向第三阶段过渡的SaaS赋予了新的内涵。云计算具有让IT能力如水和电一样按需使用的特性，这种特性加强了SaaS作为一种软件交付模式的灵活性；云计算中让IT资源由第三方提供的模式进一步降低了SaaS提供者和使用者所需负担的成本；由于云计算能够实现资源的聚集和服务的平台化，这也加快了SaaS应用间的整合步伐。下面具体讨论软件即服务层中的SaaS平台和云应用。

10.2 支撑平台

10.2.1 支撑平台的架构

软件即服务层应用类型多样，功能各异，实现方式也各不相同。提供SaaS服务的应用架构由应用类型、服务用户的数量、对资源的消耗等因素决定。一般来说，SaaS应用架构可以有四种方式，如图10.2所示。

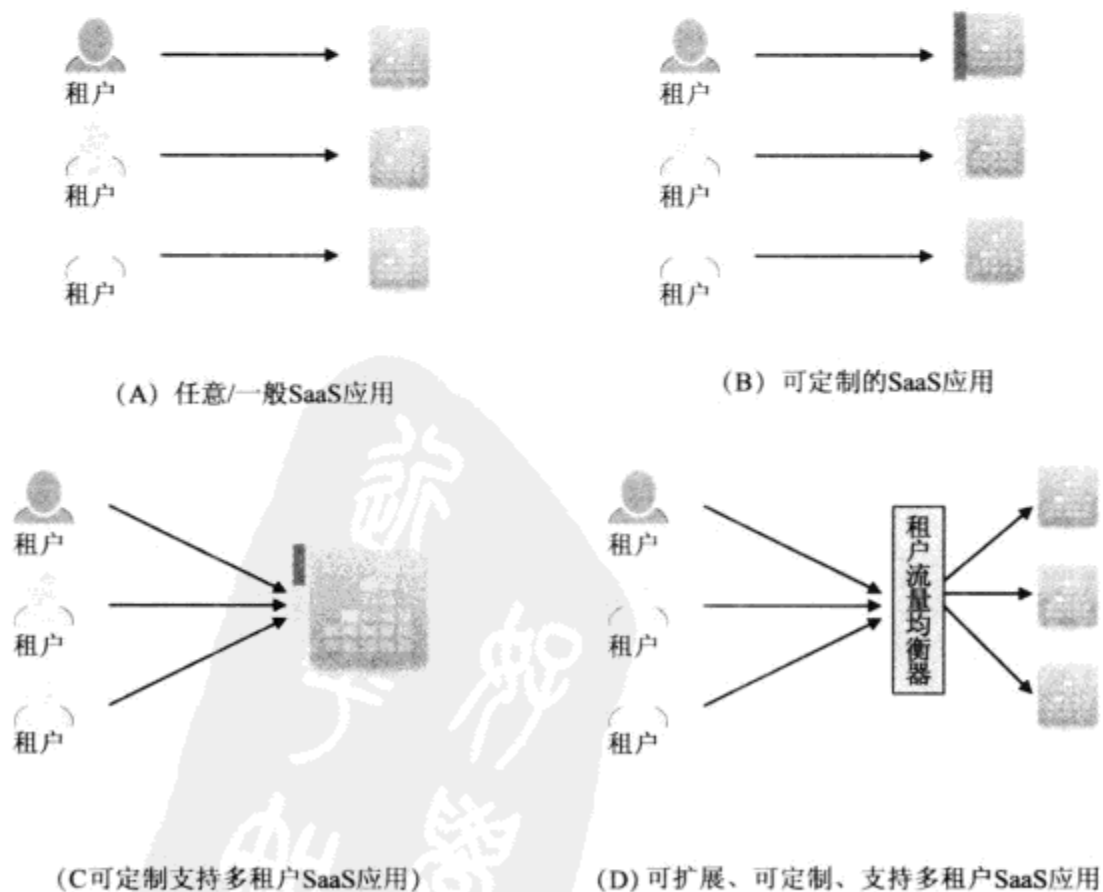


图10.2 软件即服务层的架构类型

这四种方式由是否支持可定制、可扩展和多租户三个方面的不同组合而决定。一般而言，同时支持三个方面表明应用的灵活性和可用性更强，因而更成熟，所以，如图10.2所示的从（A）到（D）四种架构也显示出不同的成熟程度，图10.2也被称为SaaS应用成熟度模型。需要指明的是，根据应用的类型不同，选择合适的架构即可，而不必追求最为成熟的架构。

图10.2中的架构（A）是一种最简单的提供SaaS服务的方式。不同租户使用SaaS平台所提供的或者迁移到SaaS平台的应用实例各不相同，即使是同一个应用也将为不同的租户运行不同的实例，每一个实例都有其不同的配置和实现。这种架构适用于快速开发的小众应用，而在开发的过程中没有过多考虑可定制、可扩展等因素。

为了增强应用的可定制性，以实现应用功能的共享，可以将应用中可配置的点抽取出来，通过配置文件或者接口的方式开放出来。当一个租户需要这样的应用时，提供者可以修改配置，定制成租户所需要的样式。在运行的时候，提供者对每一个租户运行一个应用实例，而不同租户的应用实例共享同样的代码，仅在配置元数据方面不同。这种情形就是如图10.2所示的架构（B）。架构（B）适用于那些被多次使用但使用者对于与其他租户共享实例和数据存储存在担忧的应用。例如用户希望自己的数据与其他租户的数据在存储上是隔离的，自己使用的应用服务性能不受其他租户负载的影响，或者需要遵循的法规要求如此等。

在架构（B）的基础上，不同租户可以进一步共享应用的运行实例，这就是如图10.2所示的架构（C）。在架构（C）中，每一个租户都有一套自己的特定配置，不同租户所访问的应用看起来适应自身特定所需，与其他租户的应用是不同的。但实际上，这些租户所访问的应用是同一个运行实例，但是它通过多租户技术实现了用户的配置、数据存储等方面的隔离。与架构（B）相比，架构（C）中不同租户共享运行实例需要通过可靠的多租户技术来消除用户对于运行性能和数据安全的担忧。

如果使用SaaS应用的租户数量很多，或者每个租户的工作负载起伏不定，为了有效服务用户的请求，我们希望SaaS应用不仅可定制、支持多租户，而且还应该是可扩展的。也就是说，SaaS应用的运行实例运行时所使用的下层资源与当前的工作负载相适应，运行实例的规模随工作负载的变化而动态伸缩。图10.2中的架构（D）即是这种情形。在架构（D）中，运行实例的规模可以动态变化，运行实例的前端有一个租户流量均衡器。该流

量均衡器除了具有通常流量均衡器平衡流量的功能外，还需了解服务请求所属的租户，按照租户的不同而实现服务请求的流量聚合和派发，从而实现在租户粒度上的SLA管理。在租户流量均衡器的后端是应用的运行实例。

由于SaaS应用大多是通过Web方式访问的，为了实现可扩展性，应用的架构可以采用J2EE应用模式的三层架构，即前端是处理HTTP请求的HTTP服务器，中间是处理应用逻辑的应用服务器，而后端是实现数据存储和交换的数据库服务器。三层架构的Web应用实现了传输协议、应用逻辑和数据的分离，每一层次所需的下层资源可以灵活伸缩，从而实现了整个应用的可伸缩性。

SaaS应用的开发者可以利用各种各样已有的软件组件和框架。前面所述的Web应用的三层架构已有多种软件框架支持，而开发SaaS应用可以采纳的另一种架构形式就是面向服务的架构（SOA）。在SOA架构下，SaaS应用之间可以实现互相通信：一个SaaS应用可以作为服务提供者通过接口将数据或功能暴露给其他的应用；也可以作为服务的请求者从其他应用获得数据和功能。在SaaS平台中存在大量SaaS应用的情况下，SOA可以使开发者利用已有应用，方便快捷地开发和生成新的应用。

10.2.2 支撑平台的关键技术

为了实现以上介绍的SaaS应用结构，SaaS开发者需要设计实现一系列的功能特性，以提供诸如多租户、可扩展、可整合、信息安全、计费与审计等能力，而这些功能组成了软件即服务层的关键技术集。虽然PaaS层在应用运行环境的层面向上提供了一系列的保证和支持，但是在应用本身的层面上，仍需要进行具体的工作来为软件的最终使用者提供优质的用户体验。本节首先介绍软件即服务层的设计要点，然后围绕设计要点介绍该层的关键技术。

10.2.2.1 设计要点

如图10.3所示，软件即服务层构建在硬件资源（如计算、存储和网络）及软件资源（如操作系统和中间件）上，为最终使用者提供具体的应用功能。其中，硬件资源和软件资源可以由SaaS应用提供商自己建设和维护，也可以基于本书前面章节所介绍的云中的IaaS和PaaS。

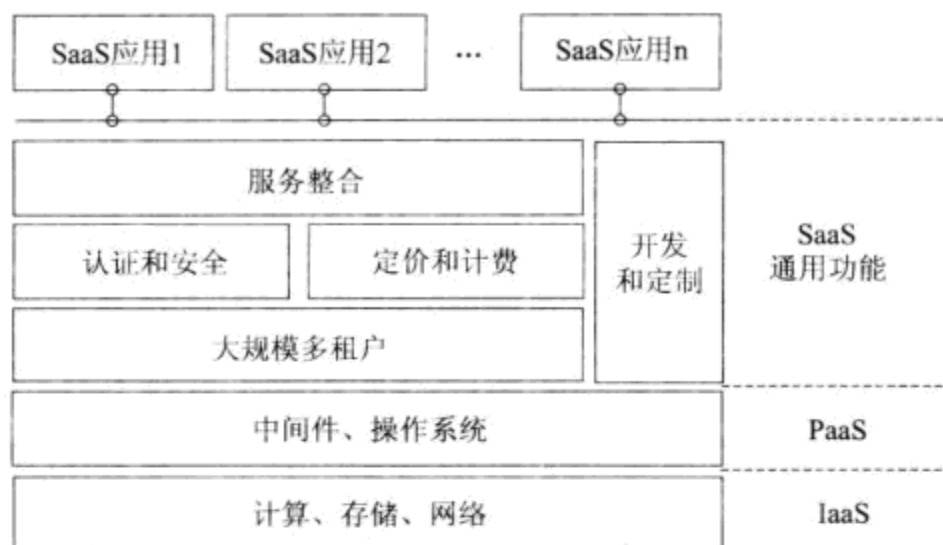


图10.3 SaaS平台的设计要点

在云计算的层次架构体系中，各个层次有其不同的分工和职能。IaaS负责提供基础设施资源，包括计算资源、数据存储资源和网络连通，并保证这些资源的可用性。PaaS负责软件运行环境的部署和维护，提供性能优化和动态扩展。各层自下而上隐藏实现细节，提供功能服务。作为最接近应用使用者的SaaS，在承接了由下面层次提供的功能的情况下，仍需要在设计上关注以下要点：

1. 大规模多租户支持

这是SaaS模式成为可能的基础。由于SaaS改变了传统应用用户购买许可证、本地安装复本、自行运行和维护的使用模式，向在线订阅、按需付费、无须维护的模式发展，这就自然要求运行在应用提供者或者平台运营者端的SaaS能够同时服务于多个组织和使用者，而多租户技术是使该需求成为可能的基础。

2. 认证和安全

这是多租户的必要条件，它改变了以往资源非共享、数据自有的应用运行模式。当应用操作请求到来时，其发起者的身份需要被认证，其操作的安全性需要被监控。虽然诸如数据与环境隔离等基础功能是由多租户技术本身保证的，但是作为应用的前端，认证和安全仍是SaaS安全的第一道防线。

3. 定价和计费

这是SaaS模式的客观要求，由于SaaS直接服务最终消费者，具有服务对象分散、需求多样、选择多的特点，因此一组合理、灵活、具体而便于用户选择的定价策略成为SaaS成功的关键。此外，由于SaaS较多采用在线订阅的方式进行购买，如何将SaaS的定价以一种

清晰、直观而便于用户理解的方式呈现也至关重要。而计费是保证整个生态系统能够良性运转和发展的最关键经济环节，也需要技术层面的有力支持。

4. 服务整合

这是SaaS模式长期发展的动力，由于SaaS应用提供商通常规模较小，难以独立提供用户尤其是商业用户所需要的完整产品线，因此需要依靠与其他产品进行整合来提供整套解决方案。这种整合包括两种类型，第一种是与用户现有的应用进行整合；第二种是与其他SaaS应用进行整合。只有通过整合和共同发展，才能营造云中良好的SaaS生态系统。

5. 开发和定制

这是服务整合的内在需要，虽然SaaS所提供的已经是完备的软件功能，但是为了便于与其他软件产品进行整合，SaaS应用需要具有一定的二次开发功能，如公开API和提供沙盒、脚本运行环境等。此外，为了应对来自上层不同应用的需求和来自下层不同运行环境的约束，SaaS应具有可定制的能力来适应这些因素。

如图10.3所示，以上五个要点是设计软件即服务层时所必须面对的，本书将其称为SaaS的通用功能。这些功能可以由SaaS应用提供商自行实现，也可以通过细化分工由专业的SaaS平台商提供，使应用商可以专注于用户需求的实现。以下各节将对多租户支持、认证和安全、定价和计费、服务整合、开发和定制这五部分依次进行深入讨论。

10.2.2.2 大规模多租户

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来降低每个企业用户的成本，同时通过支持大量的企业租户来取得长尾效应。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，该技术使得大量的租户能够共享同一堆栈的软、硬件资源，每个租户能够按需使用资源，能够对软件服务进行客户化配置，而且不影响其他租户的使用。这里，每一个租户代表一个企业，租户内部有多个用户。在多租户作为一

项平台技术时，需要考虑提供一层抽象层，将原来需要在应用中考虑的多租户技术问题，抽象到平台级别来支持，需要考虑的方面包括安全隔离、可定制性、异构服务质量、可扩展性，以及编程透明性等。同时在支持各个方面时需要考虑到应用在各个层面（用户界面、业务逻辑、数据）可能涉及的各种资源。各个层面对应用中可能涉及的各种资源提供相应的平台级多租户技术的支持，包括：

- ▶ 在持久化数据层面，包括的资源有数据库服务器、文件和IO、LDAP服务器等；
- ▶ 在业务逻辑层面，包括身份认证/授权、全局对象、远程访问服务、EJB、缓存等；
- ▶ 在 workflow 层面，包括业务流程模版、人工任务、业务规则等；
- ▶ 在用户界面层面，包括JSP及Servlet中的全局状态，以及CSS等需要定制的部分等。

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，如果每一个用户的应用都运行在单独的虚拟机上，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。

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，从而满足更特定的需求。回顾10.2.1节所介绍的软件即服务层架构，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，这样做的好处是租户间数据充分隔离，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，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。

10.2.2.3 认证和安全

图10.4展示了软件即服务层认证和安全模块的设计要点。首先，向SaaS发起的应用请求可能来自于不同的实体，如用户使用的掌上便携设备、桌面或膝上电脑，以及云中的其他应用的调用。针对这种差异化的请求，该模块需要具有前端响应处理能力，通过不同的方式来响应来自不同实体的请求。所谓不同的方式，主要是指针对访问实体的属性采用不同的认证方式。

值得注意的是，差异化的认证方式需要配合预定义的终端策略来完成。例如，对于来自便携手持设备的请求，将采用用户输入密码的方式进行认证；对于来自具有生物信息识别能力的设备的请求，将采用用户扫描指纹等方式进行认证；对于来自云中其他应用的请求，通过核对用户令牌或通行证的方式进行认证。前端响应模块根据不同的认证方式，渲染登录界面，准备接收用户输入。

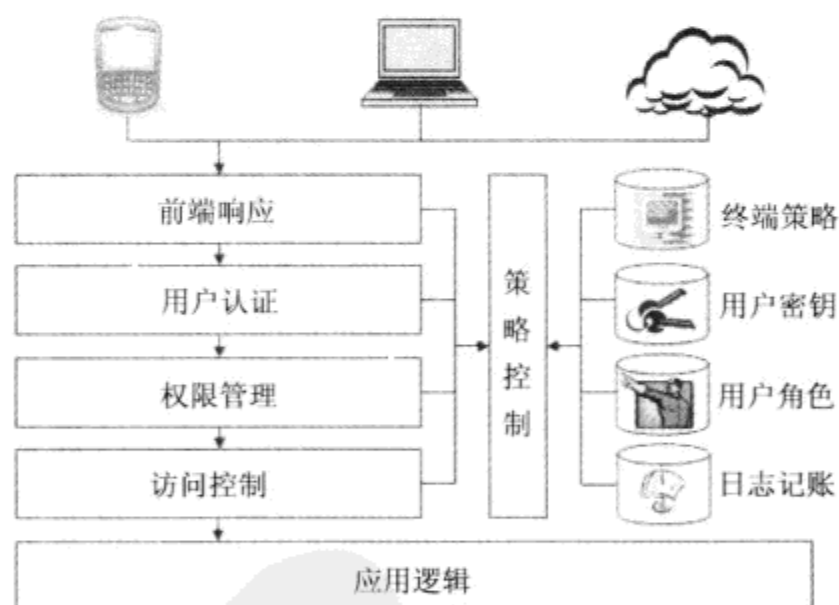


图10.4 软件即服务层的认证和安全

当用户输入登录信息后，认证和安全模块需要对用户的合法性进行确认，并且核对该用户的身份，赋予其合法的权限。这个过程需要用户认证和权限管理相互配合来完成。用户认证通过核对密钥来确认用户合法性，进而权限管理查阅用户角色目录来确定其所能访问的服务和数据。最后，当用户的身份和角色都已确定后，访问控制模块将用户请求路由至目标应用，并在该会话建立和销毁的整个过程中监控访问情况，隔离潜在的恶意行为。

用户的登录、访问和应用使用行为需要被记录下来，这就是日志记账模块的主要功能。按策略维护访问日志和用户行为不仅仅是一个常规的记录行为，更可以为其他系统模块所用。在安全层面，这些数据可以被入侵检测系统利用，通过筛选行为规则来强化安全保护；在性能方面，这些数据可以被业务系统利用，通过分析关键节点的分布来优化业务逻辑；在系统方面，这些数据可以作为用户计费的依据，作为以下将要介绍的定价和计费模块的输入。

10.2.2.4 定价和计费

在直接面对最终消费者的软件即服务层，服务定价策略的设计至关重要。不同于本书前面章节所介绍的基础设施层定价，软件即服务层的计费对象是一项具体而细致的功能，而非易于量化和描述的底层资源。因此，软件即服务层定价策略的制定需要综合考虑以下两点因素：

1. SaaS应用的核心价值

一个成功的SaaS应用提供商往往能针对用户独特而必需的需求设计出具有针对性的功能，而为了将该功能有效地交付给用户，需要一系列其他辅助功能的配合。通常，用户也可以通过其他途径获得这些辅助功能。因此，一项SaaS解决方案的价格应该依照其为用户提供价值的多少，而非为用户提供功能数量的多少来衡量。

2. SaaS应用的定价体系要做到清晰、灵活、便于理解和便于选择

其中清晰是指该应用能够提供怎样的功能，每项功能怎样计费应让使用者一目了然；灵活是指功能的不同组合或使用情况应如实反映在价格和费用里；便于理解是指定价策略应该具有清晰的框架，避免造成用户的误解；便于选择是指能够为不同类型、不同需求的用户提供针对各自实际情况的选项。

定价策略的制定直接关系到用户体验和满意程度，同时也影响着SaaS应用提供商的收益。一个好的定价策略能够促进应用提供商与消费者的有效沟通，帮助用户在互联网中快速寻找符合预期与预算的服务，帮助应用提供商提高用户忠诚度与黏性。下面将描述一个SaaS应用定价参考模型，来帮助读者理解SaaS应用的定价方法，设计符合以上所述因素的定价策略。

图10.5展示了一个SaaS应用的定价参考模型。为了达到定价的灵活性和可选择性，该

模型按照三个不同层次计费，由下向上分别是按功能、按计划（套餐）、按账户。其中，按功能付费的计费对象是SaaS应用所提供的一项功能或一组功能的集合，而其计费依据则是对该功能的使用情况。例如一个在线图片处理应用中的一款滤镜就可以作为一个计费对象，而对其的使用次数可以作为计费依据。

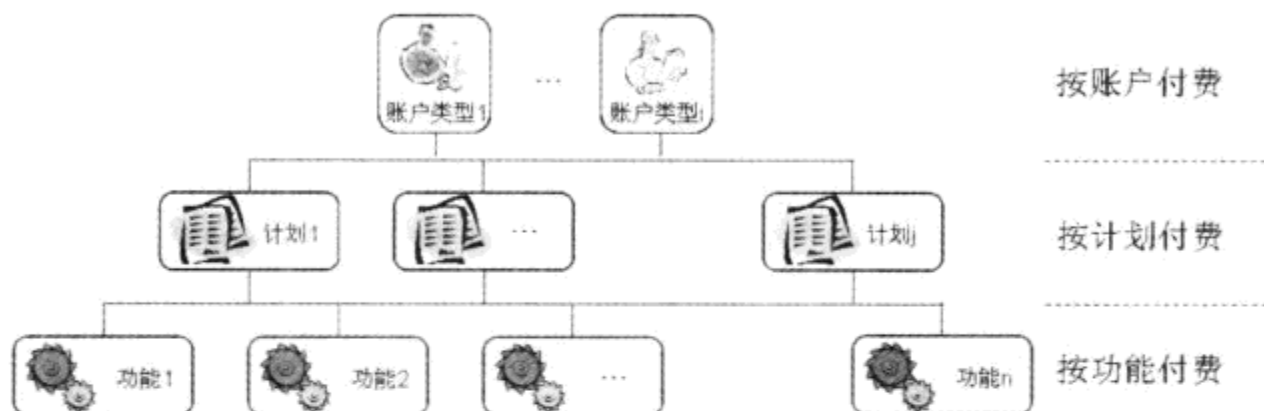


图10.5 软件即服务层的定价与计费

按功能计费虽然具有最高的灵活性，但是有其分散零碎、不便管理的劣势，也不利于提高用户的黏性。因此，在按功能计费的层次之上是按计划付费。一个计划中可以包含一个或若干个功能，以及对该功能的使用情况。与按功能计费不同，计划中加入了时间的概念，同时也可以通过差异化来细分市场，便于用户选择。例如上述在线图片处理应用可以提供两个计划，计划A允许用户在一个月中无限次地使用基本滤镜和特效滤镜，计划B则仅允许用户无限次地使用基本滤镜，特效滤镜仍需按次付费。

按账户计费的灵活性最小，但是能为用户提供最便捷的一揽子解决方案。一个账户往往是多个计划的使用者，根据账户的不同需求而进行多种计划的组合。例如上述在线图片处理应用不仅提供有关滤镜的计划，还提供有关笔刷的计划，那么就可以细分两种账户类型。全能账户A可以同时使用有关滤镜和笔刷的计划，而渲染账户B则仅使用有关滤镜的计划。

可见，在这个定价参考模型中，随着层次的提高，用户选择的灵活性越小，但其选择的便捷程度越高，对应用的使用黏性也越高。因此，SaaS应用提供商可以参考以上定价模型，选择合适的层次，在合适的时机进行定价。在应用上线的初期，为了提高知名度，提供商可以采用按功能付费的定价策略。随着应用的成熟和用户的增多，提供商可以逐渐提高定价策略的层次，并在更高的层次设置一定的价格折扣或功能增强，来吸引用户向高层次发展。这样SaaS应用才可以走上用户稳定、不断发展的良性道路。

10.2.2.5 服务整合

回顾本章开始所介绍SaaS的发展历程可以看到，从早期独立而封闭的应用到与企业现有数据和流程的深度整合，SaaS的发展伴随着其整合能力的提高。也许“整合”二字还有些抽象，我们通过一个例子来理解其中的含义。考虑本书前面章节介绍过的Salesforce CRM，它可以帮助企业自动化从营销到签单的销售环节，并为现有客户提供服务。这套系统需要能够获得企业财务系统中的销售数据，以及企业资源计划（ERP）系统中的订单数据。因此，一个SaaS应用需要与其他应用一同配合，才能够完成既定工作。在这里，整合的对象既有可能是企业现有IT系统中的应用，也有可能是企业订阅的其他SaaS应用。

如图10.6所示，服务整合自上而下针对三个层次。

（1）界面的整合。作为应用的前端，整合界面将来自于不同应用的数据、信息组合在一起，以一种自然的方式展现在用户面前，不至于带来割裂感和陌生感。

（2）流程的整合。作为应用的逻辑，整合的流程不仅要求能够沟通各个业务环节，还应具有一定的灵活可变性，使流程能够根据实际情况进行动态调整。

（3）数据的整合。作为应用的基础，数据整合需要对已有的业务数据进行验证、整理和必要的转换，使它们能够在不同的应用间传递。

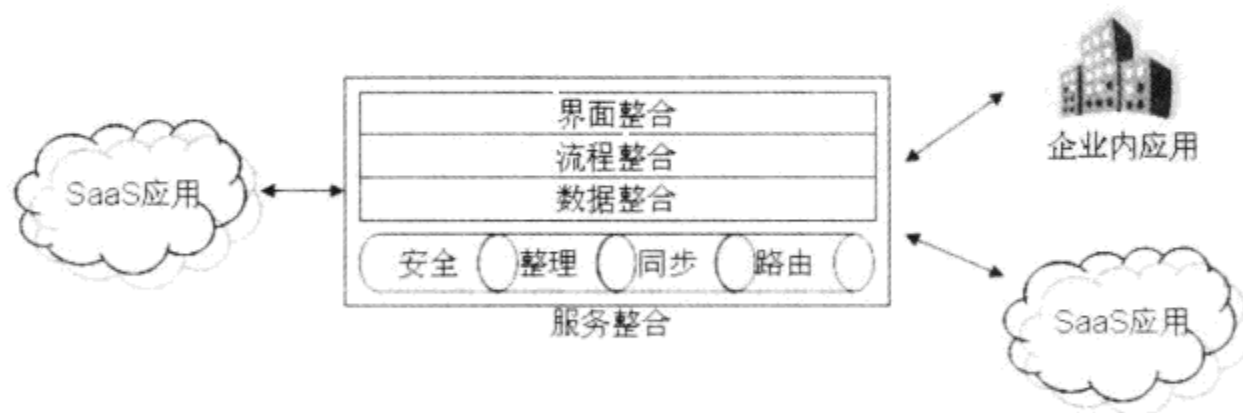


图10.6 服务整合

数据的传递是服务整合的关键，这个过程在逻辑上通常以管道的方式实现，如图10.6所示。数据在管道中流动，管道的不同部分对数据进行管理和加工。管道的长短和功能的组合由数据的特性来决定，不过以下四个部分往往是不可或缺的，它们是数据安全、数据整理、数据同步和数据路由。

- ▶ 数据安全模块负责对进入管道的数据进行来源认证，完整性检查，保证数据是可信而未被篡改的。该模块还可以对数据进行加密，并辅助进行访问控制和病毒防疫。
- ▶ 数据整理模块负责对数据的格式进行识别，剔除重复、过时或不符合要求的数据，或者将问题数据进行格式转化。该模块还可以组合多个不同来源的数据，以辅助业务逻辑的整合。
- ▶ 数据同步负责根据业务的规则和流程来控制数据的流动，确定数据的传递和更新次序，避免由于中间环节异常而造成的错误更新和不同步现象。
- ▶ 数据路由是管道的出口，它负责将每一份数据投递到目标应用。投递的规则可以来自数据外部，比如被识别的数据源；也可以来自数据内部，比如某一字段的具体数值。

服务整合往往是SaaS应用商所提供解决方案的一部分。整合的功能可以由应用商自行提供，也可以由第三方的专业公司提供。后一种方式逐渐成为主流，成功的云整合/服务整合服务提供商不断出现。这是因为整合的工作除了需要技术功底，如数据管理、网络传输和界面开发，更需要对被整合对象的理解和经验。专业的整合服务提供商可以在为SaaS提供者和使用者服务的同时积累这样的经验，形成现成可用的模板，加速整合的进度。可见，现今的SaaS已经形成了一套生态系统，理解这个系统并寻找自己的位置是SaaS提供商成功的基础。

10.2.2.6 开发和定制

开发和定制是SaaS平台为终端用户、独立软件提供商、服务集成商提供的第五个通用功能。开发和定制的核心技术要求是，SaaS应用能够以一种标准的、简单的方式提供开放的接口，如果可能，还需要为用户、开发者、集成者提供一个易用、安全的测试环境。下面分别介绍开放接口技术和测试环境技术。

开放接口技术是伴随着互联网的发展而逐渐被各种用户接纳的技术。最先是由Amazon、eBay等电子商务网站提供了针对网站上商品信息查询的开放接口，目的是使用户通过更多途径访问网站，以及进行二次开发。随后Yahoo、Google等搜索引擎也提供了开放的搜索和查询接口，而几年前Google Maps开放接口的推出，使得大量基于地理位置的第三方定制应用成为可

能。目前主流的电子商务网站如Amazon、eBay、淘宝，社会化交互网站如Facebook、Twitter、开心网、新浪微博、腾讯微博、搜狐微博等已经把提供开放接口作为必需的功能。到目前阶段，开放接口技术已经涉及应用业务流程的各方面，包括信息查询、状态更新、用户认证等。

目前主流的开放接口实现技术是SOAP和REST，以下分别进行介绍。

SOAP的全称是简单对象访问协议（Simple Object Access Protocol），它的初始设计目的是用于在网络上为Web服务交换结构化的数据，数据采用XML（可扩展标记语言）的形式表述。SOAP可以与HTTP、SMTP、RPC等应用层传输协议搭配使用，完成协商、消息通信、数据传递的任务。目前主流的应用服务器，从企业级的WebSphere Application Server，到开源的Apache Tomcat都对SOAP有良好的支持。一个典型的SOAP使用场景是，SaaS平台在应用服务器上提供一个SOAP的服务，用户通过客户端发送一个SOAP消息包，如果使用HTTP协议，可能是一个HTTP POST消息，在消息的主体部分就是SOAP消息，SaaS平台收到SOAP请求时，解析消息包的内容，查询本地数据库或进行相应的操作，然后把查询结果或者操作结果以SOAP消息包的格式返回给用户，在使用HTTP协议时，这个消息会封装在一个HTTP RESPONSE消息里。SOAP的优点是它可以与很多现有传输协议搭配工作，易于推广，但是由于它的设计动机是为了Web服务，而Web服务的需求使得SOAP需要做很多高级的扩展，导致SOAP的学习难度较高，操作也相对复杂，性能会受到影响。

REST的全称是表述性状态转移（Representational State Transfer），它是一种针对网络、分布式应用的软件架构理念和风格。REST最早由Roy Thomas Fielding在他的博士论文Architectural Styles and the Design of Network-based Software Architectures中提出。经过发展演化，REST目前确定了几条约束规则：

（1）客户端与服务器分离开发，通过接口交互，服务器端专注数据处理和业务逻辑，客户端专注用户界面和用户状态，并保证移植性；

（2）无状态，在通话中，服务器端不维护会话的状态，客户端负责维护状态，若服务器端需要暴露状态，需要把状态作为一个资源通过URL暴露出来；

（3）可缓存的，在互联网中客户端会缓存服务器的应答信息，应答信息的设计应该保证能够被正确缓存；

(4) 层次化系统, 一个客户端不能确定它是在和最终层次的服务器通话, 还是在和一个层次中某层的服务器通话, 这样保证了系统的安全性和可扩展性;

(5) 一致的接口, 通过一致的接口保证客户端与服务器的相对独立, 具体来说, 服务器的每一个资源都会有一个唯一的资源标识 (Uniform Resource Identifier, 简称URI), 客户端通过访问URI对资源进行操作, 而访问所包含的信息包能够描述信息如何被处理及返回。

只要遵守以上五条约束规则, 就可以称为RESTful的设计, 而具体实现可以由开发者自己决定。目前比较典型的REST使用场景是与HTTP协议结合, 资源暴露为URL, 客户端通过HTTP定义的四种操作原语GET、PUT、POST、DELETE对资源进行查询、增加、修改、删除操作, 资源的内容会放在HTTP报文的报文体 (body) 部分, 并支持XML和JSON两种格式。

从上面的描述可以看出, REST的规范清晰, 学习、使用起来也比较简单, 开发者只需了解HTTP、XML、JSON等基础知识即可进行开发, 并且由于REST实际上规范了资源的查询、修改、添加、删除操作的接口名称就是HTTP的四种操作, 这样大大增加了开放接口的通用性, 开发者不需再阅读大量由接口提供者撰写的不通用的接口文档。目前, 提供REST风格的接口几乎已经成了所有服务提供者的共识, 在工程上, 服务器端只需要在应用服务器上增加Restlet、Apache Wink等扩展包就可以支持REST。

与开放接口技术同等重要的定制与开发相关技术是测试环境, 具体来说称为沙盒 (sandbox)。沙盒是一个隔离的测试环境, 它可以模拟生产环境、实际系统的状况, 开发者可以在沙盒里测试代码, 寻找代码的功能问题和性能问题, 而不会影响到实际系统的功能和数据。沙盒可以有不同的模拟级别, 例如它可以只模拟实际系统的最小功能集, 也可以模拟出实际系统的软硬件环境, 甚至提供与实际系统类似的数据集或者数据库。当然模拟的级别越高, 实现成本也越大, 在具体使用中, 沙盒的模拟级别可以对应代码开发的阶段, 在功能测试时使用最简环境, 而在上线前的最终测试时使用模拟了数据库和软硬件环境的环境。此外, 沙盒还需要能够从技术上支持测试代码向生产环境的迁移, 例如需要内嵌对于代码版本控制 (CVS、SVN等)、开发测试文档、日志等的支持。

10.2.3 支撑平台的参考实现

综合上文所介绍的设计要点和关键技术, 本节给出一个SaaS平台的参考实现架构, 如图10.7所示。值得注意的是, 该参考架构的目标实现者是SaaS平台提供商。回顾本章开始

所描述的SaaS生态系统，SaaS平台的作用是为SaaS软件开发者（ISV）提供应用所需的通用功能部件。

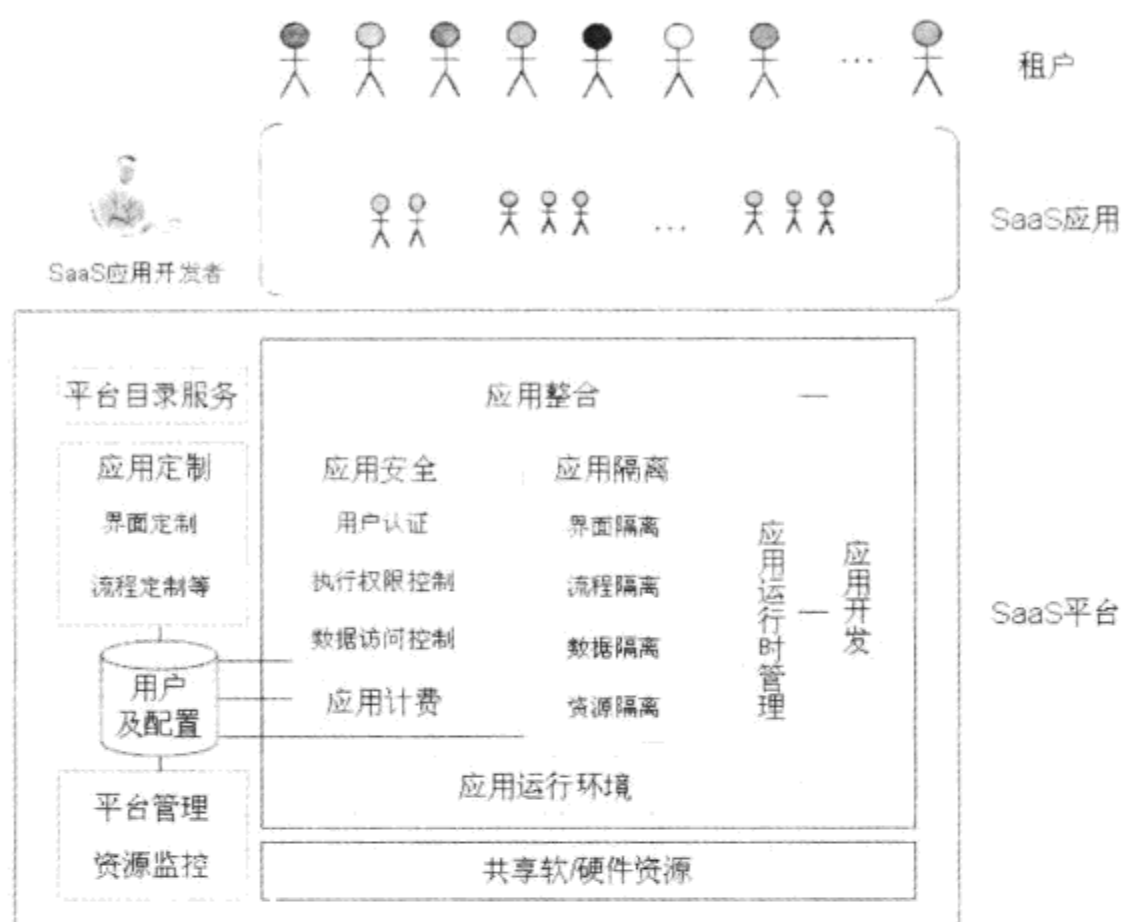


图10.7 SaaS平台的参考实现架构

在图10.7中我们可以看到应用安全、应用计费、应用整合、应用隔离等功能部件。它们对应了上文所介绍的关键技术部分，在此不再复述。此外，应用定制、应用隔离等功能部件是实现多租户的基石。值得注意的是，应用定制除了针对界面进行，还能够针对流程等方面进行更加深入的定制。同样，应用隔离除了数据隔离，位于其上的界面隔离和流程隔离以及位于其下的资源隔离，都是该部件所具有的能力。由此可见，该平台能够为应用开发者提供较强的功能性支持，使他们可以专注于业务的开发。

除此之外，该平台还具有应用的运行环境，并且能够对其进行运行时管理。该部件负责沟通底层的共享软件和硬件资源，使它们能够为应用所用；此外它还提供应用的上线、运行时管理、离线维护和下线等功能。配合应用隔离部件和应用开发部件，应用运行环境能够实现对应用的能力管理。该服务使应用开发商能够根据对业务负载的考虑来选择对资源的消耗。这里的资源既包括共享的软、硬件资源，也包括对SaaS平台功能的使用。例如，应用运行环境具有数据缓存的能力，它在数据持久化层和业务逻辑层之间加入了缓存，通过提高数据的读写效率来提高应用的性能。应用开发商可以根据其

自身情况来决定是否需要该功能。能力管理服务将根据用户的选择来保证其对资源的合理使用，并通过应用计费部件向应用开发商综合收费。

除了以上针对SaaS应用所必需的平台功能外，该参考实现架构中还提供了诸如平台目录服务等为应用开发商提供的增值服务。应用开发商可以把其开发的应用产品注册进入平台目录，由SaaS平台统一负责推广。如果SaaS平台具有较大的影响力和较好的声誉，这无疑将为应用的流行提供有利条件。此外，SaaS平台也应具有对其本身的运行进行监控和管理的能力。在这里，对底层资源消耗的监控尤其重要，尤其是在SaaS平台本身不维护底层资源，而依靠云中其他平台提供服务的场景下。

10.3 云应用

10.3.1 云应用的特征

(1) 云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，软件即服务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。

(2) 用户在使用云应用时，不需要进行先期投入，只需要在使用的过程中按照实际的使用情况付费。首先，用户在使用云应用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，但用户不需要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。

(3) 软件即服务层的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大为降低。由于应用都是运行在云中而且彼此相对独立的，因此云应用整合较传统应用会相对容易实现。

10.3.2 云应用的分类

我们知道，云应用是运行在云端应用的集合。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，并且通过服务接口与用户交互。总的来说，软件即服务层的应用可以分为三大类：

(1) 面向大众的标准应用，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程序的云上版本。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。标准应用的典型代表有Google的文档服务Google Docs、IBM的协作服务LotusLive等。

(2) 为了某个领域的客户而专门开发的客户应用，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。客户应用是传统的企业IT解决方案的云上版本。客户应用的典型代表有Salesforce的CRM应用和NetSuite的ERP应用。

(3) 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。在图10.8所示的长尾模型中，横轴是云应用按流行度的排序，纵轴是云应用的流程度。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”。

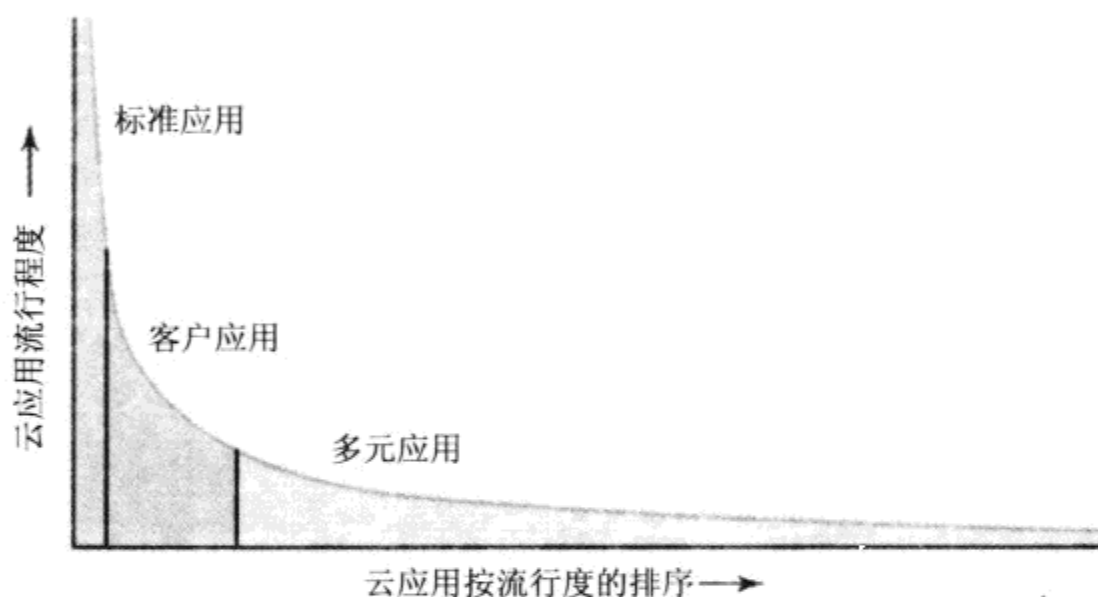


图10.8 云应用的长尾模型

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。

客户应用针对的是具有普遍性的某种需求，比如客户管理系统（CRM）和企业资源规划系统（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，都可以创造价值。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。

10.3.3 云应用的典型示例

10.3.3.1 标准应用示例

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，用户不必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，这在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。

10.3.3.2 客户应用示例

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 and 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软件即服务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，还需要具有一定的互操作性。互操作性一方面考虑如何将现有的应用迁移到云中，另一方面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服务平台的时候，就意味着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平台中导出并导入到另一个云平台中。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云计算开放标准建立，这需要当前IT公司的共同推动。

NetSuite也是一个比较成熟的客户应用，它是一个在线的企业资源计划、客户资源管理及电子商务管理系统，目前有6000多家公司在使用NetSuite。NetSuite还提供了NetSuite OneWorld应用，用以帮助企业进行全球化的实时管理，以及商业智能、数据挖掘应用SuiteAnalytics，工作流和业务流程管理应用SuiteFlow。另外NetSuite还提供了SuiteCloud业务

云平台，为用户提供运行SaaS应用的基础设施平台、应用开发平台、应用分享平台和技术支持平台。

另一个典型的客户应用是Workday，它是一个完全在线的人力资源、财务和支付服务。当前的企业在人力资源管理上面临着很多问题，比如如何有效地控制人员、如何尽量发挥表现突出人员的能力、如何了解项目的前提条件和产出，以及如何处理公司扩张和多元化带来的挑战等。Workday通过对人力资源、项目、目标、财务等数据的整合，为企业提供完整的人力资源管理解决方案，涉及员工福利、考勤管理、员工培训、员工交接、薪资管理等方面。进而Workday将服务扩展到企业的财务管理层次。

Chiquita是一个高端新鲜水果和蔬菜的生产和运输公司，在80多个国家拥有2万多名员工，员工经常出现离职情况，对企业业务的连续性造成影响，通过Workday的人力资源服务，他们对员工进行数据挖掘，发现哪些员工可能会在近期离职，并针对离职做出预见性的工作交接，从而大幅减少了因员工离职造成的业务损失及招聘费用，每年节省的费用可达200万美元。另外通过Workday的测算，他们发现，之前公司制定的将生产基地尽量建立在客户附近，从而减少运输开销的策略是不经济的，实际上，建立几个集中的基地在人力成本较低的地点会更加划算，通过Workday的帮助，他们虽然在运输开销上有一定的增加，但是降低了总体的成本。

10.3.3.3 多元应用示例

多元应用是软件即服务层中最为丰富多彩的一类应用，涉及个人、公司、团体工作生活的方方面面，并跨越了多种平台和接入设备，下面介绍几个典型的多元应用。

在传统的PC平台上，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GoogleMap的API得到使用者目前所处的街道位置，以及附近所有的公交站、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，则需通过客户端脚本（如JavaScript）或Applet来完成。

随着移动设备、智能终端（手机、平板电脑）的大范围普及使用，移动设备平台上的应用也变得越来越丰富。目前使用者最多的两个应用平台，一个是Apple公司为iPhone/iPad设备提供应用开发、使用、下载的App Store平台，一个是为Google开发的Android平台提供应用的Android Market。下面重点介绍一下App Store及其中的典型应用。

App Store（应用商店）是苹果公司于2008年推出的应用开发、上传、下载、更新、计费平台。它提供了应用的程序开发包（SDK），应用程序开发者通过使用SDK，开发自己的应用程序，并可以注册App Store的开发者账户，将自己的应用发布到App Store上，并选择该应用免费下载或者收取一定的费用才能下载。普通用户可以在平台上看到所有的应用程序，并根据自己的喜好下载到自己的电脑、iPhone、iPad上使用，如果是收费应用，用户会从自己的信用卡中支付费用给App Store，App Store将费用提取一部分管理费后，剩余部分交给开发者。

这种模式形成了一个良性的循环，保证了平台各方的多赢：应用的开发者可以自由定价，通过应用收费得到收入，保证自己的生活甚至建立创业公司；而用户可以选择自己想要的免费或者收费应用，通过各种应用获得便利；同时也正是因为App Store的成功，为Apple公司带来了稳定的平台提供收入，并直接带动了Apple公司的iPhone、iTouch、iPad等终端设备的销售，使得Apple公司成为目前发展势头最强劲的IT公司之一。根据独立网站App of the Day的统计，截至2010年12月，App Store上有6万多个开发者，提供超过30万的应用可供下载，其中三分之二为收费应用，三分之一为免费应用，应用的平均定价约2.5美元（相当于购买一盒鲜牛奶），85%的应用针对iPhone手机，7%的应用针对iPad平板电脑，8%的应用可以同时支持两种设备，应用类型涉及书籍、游戏、娱乐、教育、生活、健康、

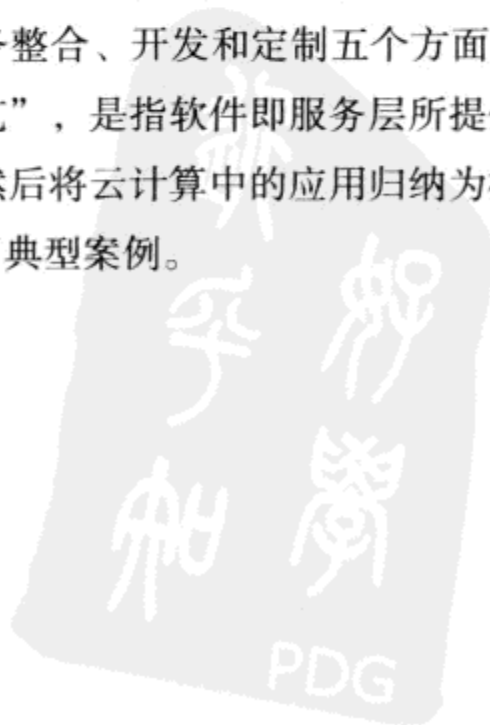
工具、旅行、办公应用等类别。根据Apple公司网站上公布的数据，App Store上应用程序的下载次数已经超过了100亿次。

FlightTrack（航班跟踪）是App Store上一个非常受欢迎的多元应用，针对的是经常坐飞机旅行的商旅人士。它可以为用户显示航班时刻表，用户可以输入自己即将乘坐或关注的航班，程序会获取网络上由航空公司、机场、空管部门提供的信息，汇总处理后显示给用户，用户可以看到航班当前的状态，是否会取消、晚点出发、晚点到达，甚至可以看到正在飞行的航班目前已飞到了哪里。用户还可以查询登机口，查看航班对应飞机的座位图，以及若航班取消，是否有合适的替代航班。FlightTrack被《华尔街日报》评价为航班状态覆盖最广的专业软件。

10.4 小结

云应用是指运行在云中、以软件即服务（SaaS）的形式提供给客户的应用，用户通过浏览器或者开放接口访问应用，按需付费，不需进行一次性投入，并通过使用整合的多种应用来提高效率、获得新创价值。

本章首先描述了SaaS生态系统及应用提供商可以选择的市场定位，回顾了SaaS的发展历程，总结了其在云计算时代的特征，再从“技”和“艺”的角度对软件即服务层进行了深入剖析。所谓“技”，是指实现SaaS平台所必须考虑的技术问题。本章首先介绍了软件即服务层的架构，然后分析了SaaS平台的设计要点和关键技术，分别从大规模多租户、认证和安全、定价和计费、服务整合、开发和定制五个方面深入展开，并给出了一个SaaS平台系统的参考实现。所谓“艺”，是指软件即服务层所提供的丰富多彩的应用服务。本章首先总结了云应用的特征，然后将云计算中的应用归纳为标准应用、客户应用和多元应用三大类，并针对每一类给出了典型案例。



第四篇

业界动态篇

- 第11章 虚拟化的业界动态
- 第12章 云计算的业界动态

「第四篇」
业界动态篇



第 11 章

虚拟化的业界动态

- 11.1 IBM
- 11.2 VMware
- 11.3 Xen/Citrix
- 11.4 KVM/Red Hat
- 11.5 Microsoft



虚拟化是实施云计算必要的前提，那么，当前掌握虚拟化核心技术的IT厂商有哪些，它们各自的发展和技术特点是什么？本章将一一为读者解答这些问题。

11.1 IBM

11.1.1 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11.1所示。采用虚拟化技术后，在一台IBM7044（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，因此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11.1 IBM 7044（M44）



从产品角度看，IBM公司从服务器等基础IT设备的虚拟化到应用的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。

（1）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。

（2）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。

（3）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。

11.1.2 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，它们都能够与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HiperSocket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。HiperSocket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。

在操作系统方面，z系统专用的操作系统提供了Workload Manager（WLM）模块。该模块能在一个z/OS系统中运行多个工作负载，并管理这些负载的优先级。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11.2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。

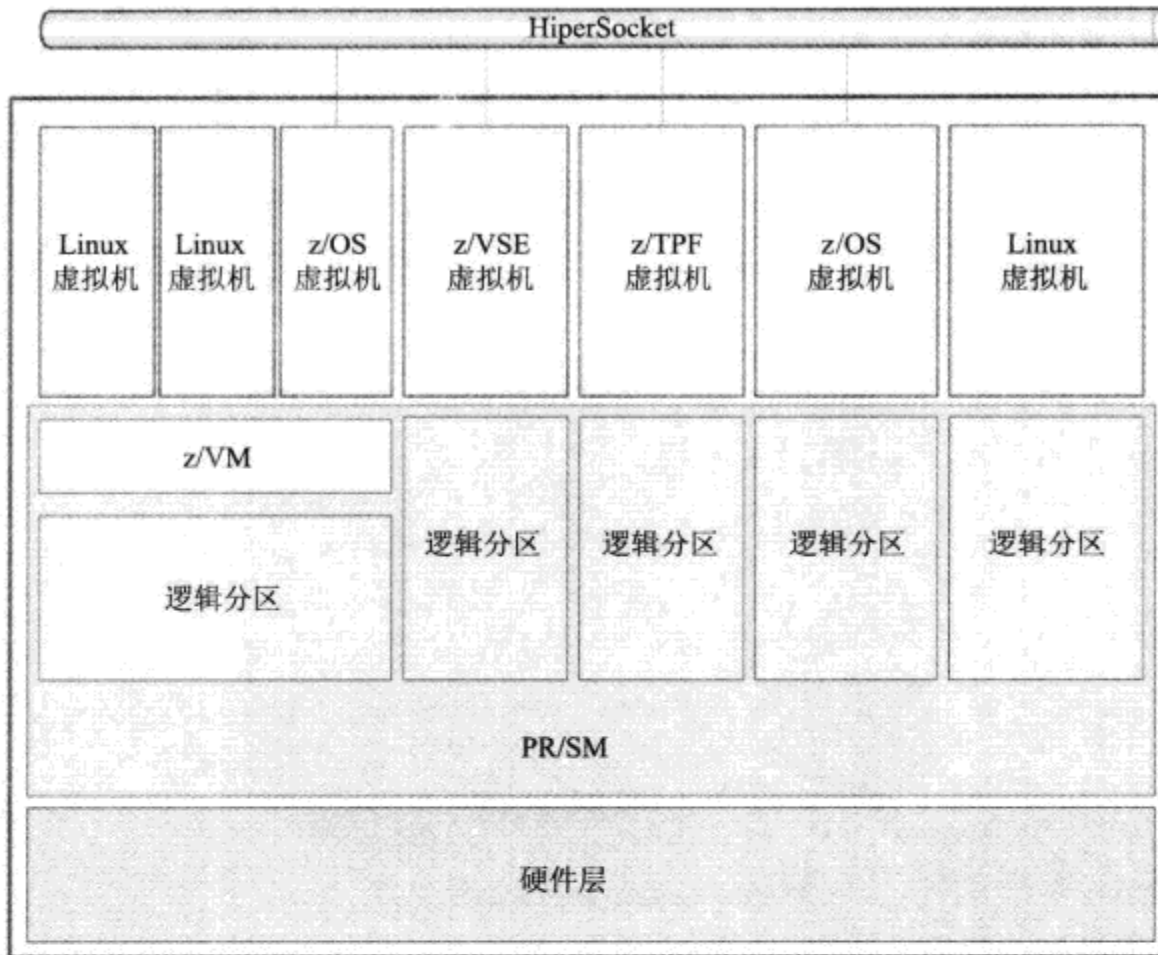


图11.2 IBM System z虚拟化架构

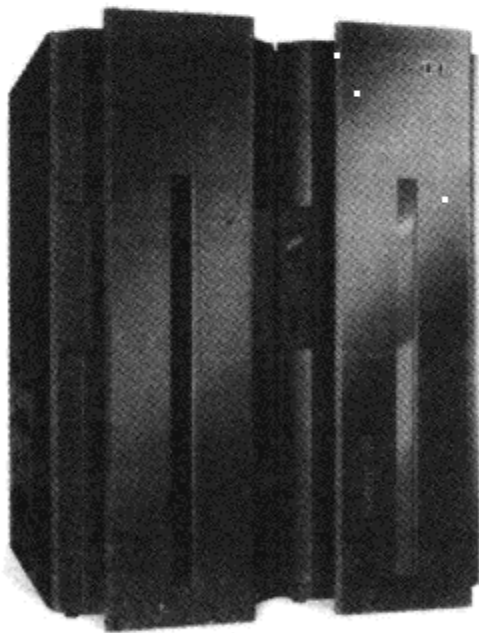


图11.3 IBM System z10服务器

总之，IBM在System z系列主机积累几十年的虚拟化经验使z系列成为承载异构软件解决方案最理想的技术平台。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11.3所示）能够同时运行上千台z/OS、z/Linux和TPF虚拟机。

11.1.3 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11.4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 workstation 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。

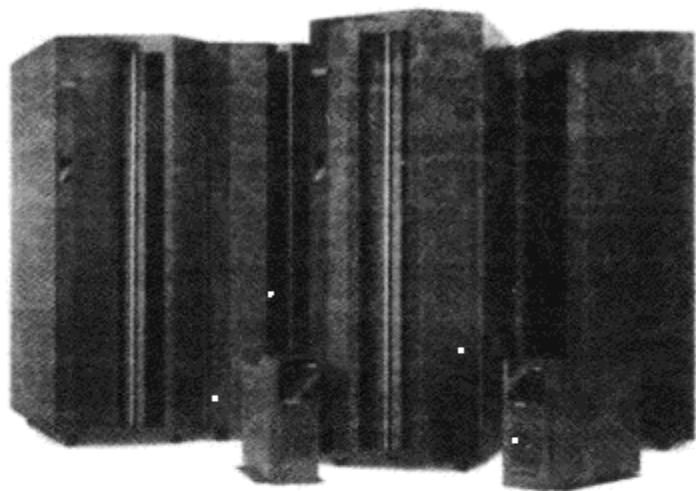


图11.4 IBM System p服务器

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, 它可以在无须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11.5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。

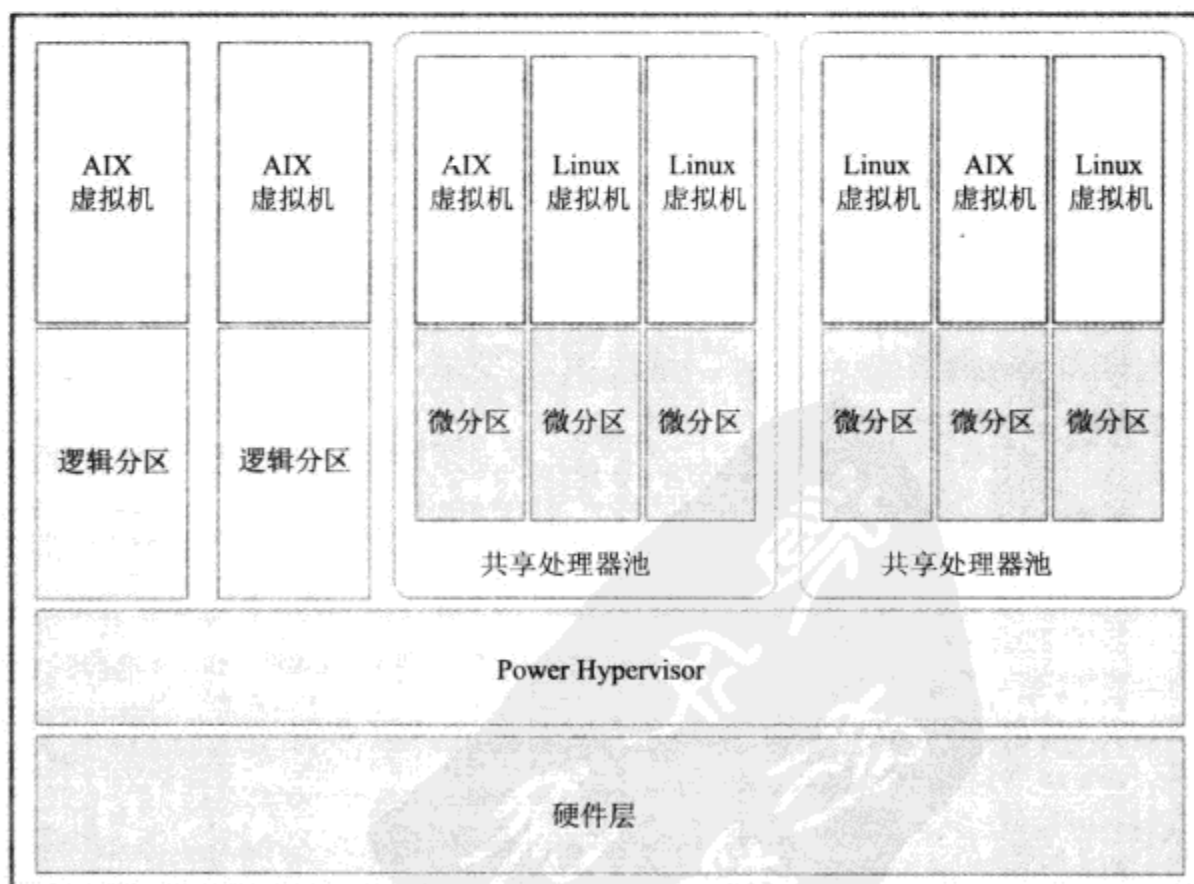


图11.5 IBM System p虚拟化架构

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。

11.1.4 虚拟化管理

本节按照第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11.6所示。通过扩展 workflow，TPM还能够实现其他管理功能，例如安装系统更新等。

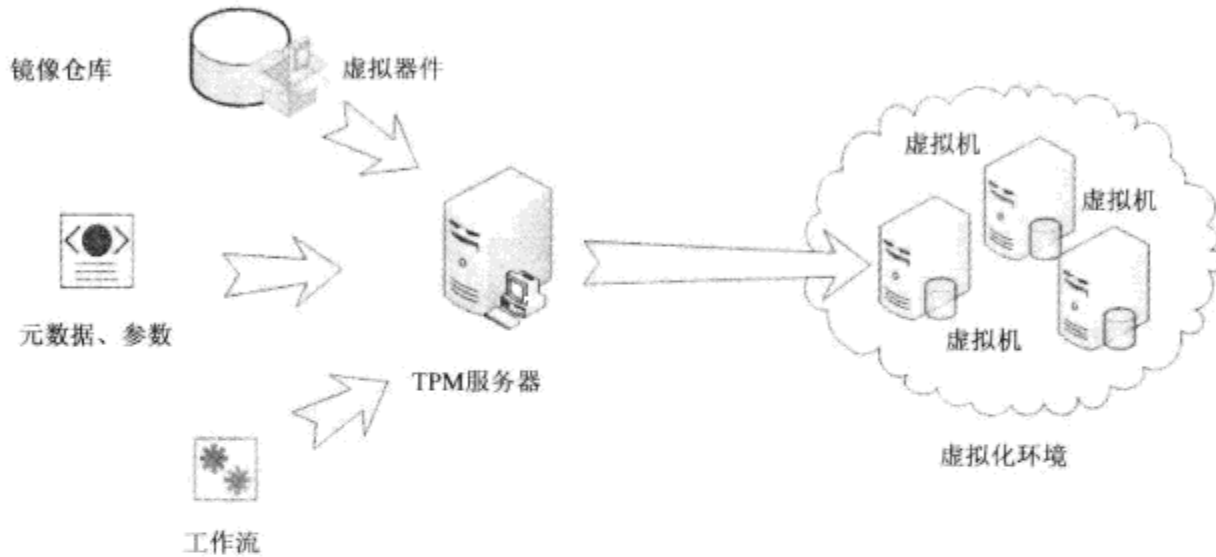


图11.6 TPM部署虚拟器件

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运营部门无须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11.7是ITM监控虚拟化环境的示意图。

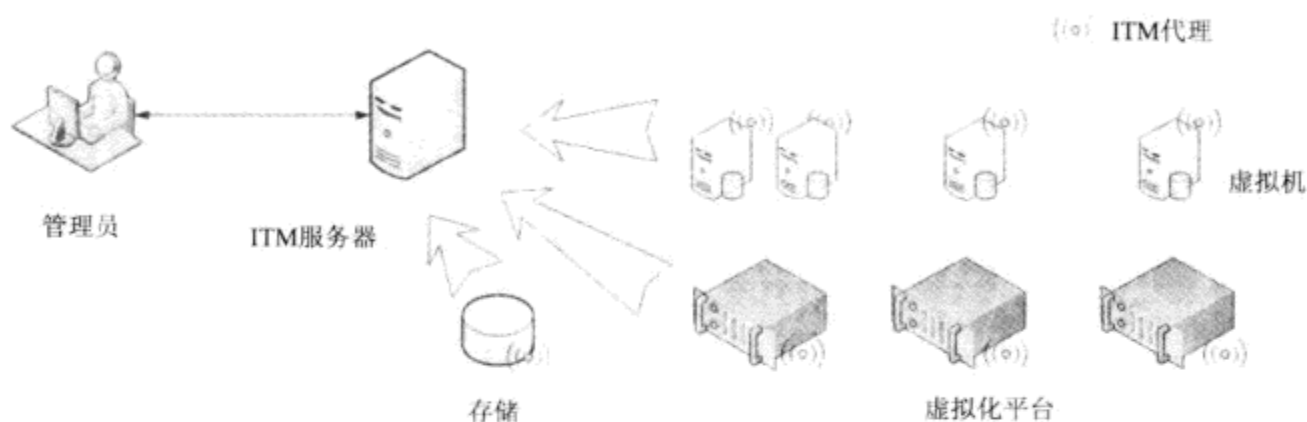


图11.7 ITM监控虚拟化环境示意图

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。

11.2 VMware

11.2.1 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。下面将分别介绍这些产品线。

11.2.2 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的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，并且还能够在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通过规范化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。

11.2.3 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Windows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须进行回归测试，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。

11.2.4 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。

11.3 Xen/Citrix

11.3.1 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，这个版本的Xen是第一个真正能够满足企业需求的版本，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，它们联合开发代号为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”的云概念，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11.8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。

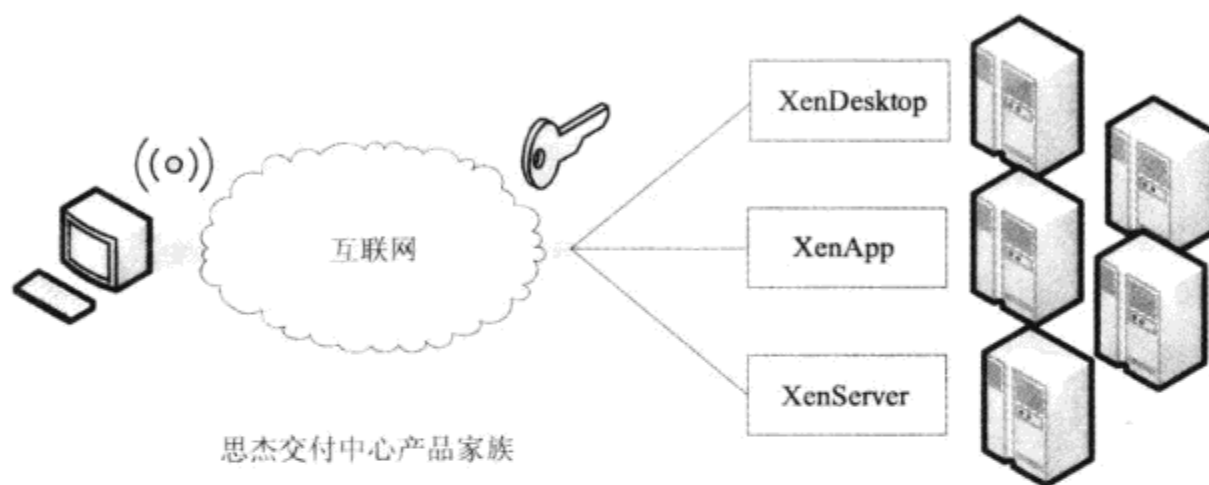


图11.8 思杰交付中心产品家族

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的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。

11.3.2 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。

11.3.3 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的服务器中，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。

11.3.4 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，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。

11.4 KVM/Red Hat

11.4.1 概述

KVM是指基于Linux内核（Kernel-based）的虚拟机（Virtual Machine），也被称为“内核虚拟机”和“K-虚拟机”。它是开源虚拟化技术的最新代表，使用Linux系统的核心调度功能进行虚拟机的管理。自Linux 2.6.20之后，KVM便集成在Linux的各个主要发行版本中，已成为主流虚拟机管理平台之一。

虚拟化技术在最近几年取得了长足的进步，软件虚拟化也一直以来是IT业界关注的焦点，包括全虚拟化和半虚拟化。全虚拟化调用处于虚拟操作系统和硬件层之间的管理程序或虚拟机监视器（Virtual Machine Monitor, VMM）来完成系统虚拟化，客户机操作系统或应用程序不需要进行任何修改即可在虚拟环境中运行；半虚拟化管理平台通过在一个协作环境中运作来实现底层硬件的虚拟化，因此需要对运行在虚拟机上的客户机操作系统进行

修改。这些客户操作系统会意识到它们运行在虚拟环境中。为了支持虚拟化技术的发展，Intel和AMD扩充了其“指令集”，对虚拟机管理平台提供了直接的硬件支持。

2005年，Intel公司发布了两款CPU产品，第一次装载了Intel Virtualization Technology（Intel VT）的虚拟化技术，使得一个CPU工作起来就像多个CPU并行运行，从而支持在一部电脑内可同时运行多个操作系统。

2006年，AMD公司推出了包含支持虚拟化的硬件辅助技术AMD Virtualization（AMD-V）的系列处理器，具备AMD-V功能的这些处理器有一套新的指令集，用于帮助虚拟监视器进行虚拟化，并有效降低了虚拟化的开销，提高了计算性能。

2006年末，在AMD-V和Intel VT虚拟化技术的发展基础上，KVM应运而生，作为一种完全虚拟化技术的解决方案，于当年10月被以色列的一个开源组织“Qumranet”提出和开发出来。

2007年初，KVM被Linux内核的总协调人采纳，随后吸收进去集成在Linux 2.6.20版本而正式对外发布。

2008年，Red Hat公司收购基于KVM进行虚拟化管理的Qumranet项目产品，集成到红帽公司的企业虚拟化产品Red Hat Enterprise Virtualization (RHEV)当中。

2009年，Red Hat公司发布企业版RHEL5.4（RHEL：Red Hat Enterprise Linux），第一次正式推出基于KVM虚拟化技术的Linux企业版操作系统。

2010年，Canonical公司发布支持KVM的Ubuntu服务器版操作系统 Ubuntu 10.04。

鉴于KVM的快速崛起，市场上陆续出现了诸多采用KVM的代表性产品，包括Red Hat的企业版 RHEL5.4及其后续版本； Canonical公司发布的Ubuntu服务器版操作系统 Ubuntu 10.04及后续版本；Novell公司发布的SLES 11 SP1及后续版本（SUSE Linux Enterprise Server）。

11.4.2 架构

在KVM的架构中，虚拟机作为正常的Linux系统进程而执行，被标准的Linux调度器进行规划管理，每一个虚拟的CPU都可看作为一个正常的Linux系统进程。图11.10和图11.11提供了KVM虚拟化架构的视图。带有KVM模块的Linux内核系统管理程序运行在Intel VT或AMD-V处理器所支持的能够进行虚拟化的硬件平台上，跟普通Linux内核类似支持运行

应用程序的同时，还可以支持通过KVM工具加载的虚拟机客户操作系统，使得主机操作系统所支持的相同应用程序能够在虚拟机中正常运行。设备仿真处理由修改后的QEMU(模拟处理器的自由开源软件)提供了一套模拟的基本输入输出系统（BIOS），PCI总线，USB总线，标准设置IDE和SCSI磁盘控制器，网卡等设备。基于KVM的整个虚拟化架构很好地支持了CPU、内存和I/O的全虚拟化。Intel VT 或 AMD-V辅助的处理器直接提供了CPU虚拟化支持，内存可以通过KVM进行虚拟化，而I/O 通过一个修改的QEMU进程进行虚拟化。

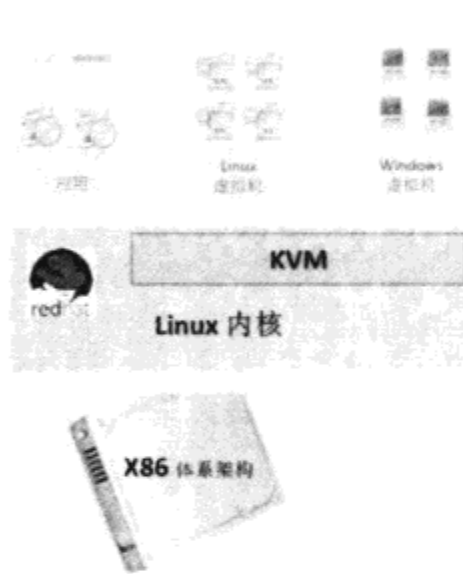


图11.10 KVM系统架构

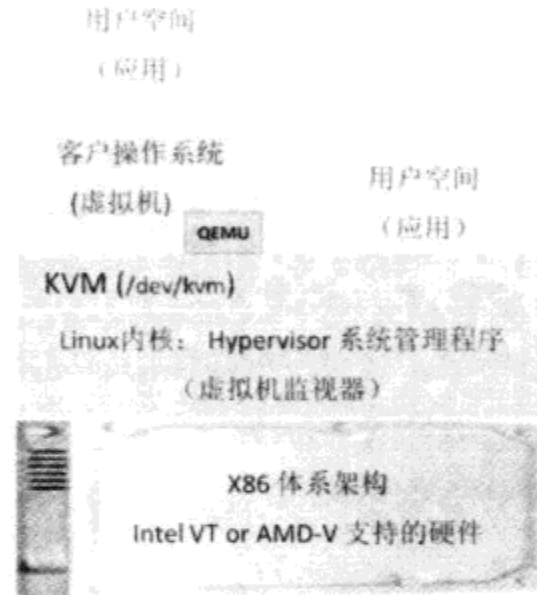


图11.11 KVM虚拟化架构

KVM 通过创建/dev/kvm设备和启用独特的客户模式实现了内存虚拟化。KVM 向 Linux 中引入了一种除现有的内核和用户模式（运行在内核外的程序）之外的新进程模式，即客户模式，用来执行非I/O操作的客户操作系统的代码。客户操作系统在客户模式中运行可以支持标准的内核，而在用户模式下运行则支持自己的内核和用户空间应用程序。KVM通过加载内核模块而将Linux内核转换为一个系统管理程序，这个内核模块导出了一个名为/dev/kvm的设备，它可以启用内核的客户模式，使得每个虚拟机客户操作系统有自己的独立于内核或其他运行中虚拟机的地址空间。该地址空间是在实例化客户操作系统时映射的，保证了访问/dev/kvm设备的进程获得与之对应的不同映射，从而确保了虚拟机之间的隔离。KVM 采取了影子页表（Shadow Page Table）机制实现客户物理地址到主机物理地址的转换，CPU 可以通过在访问未经映射的内存位置时使用系统管理程序来支持内存转换的功能。

另外，由于KVM采用的客户模式并不执行非I/O操作，在客户操作系统上执行I/O的功能是由QEMU提供的。QEMU是一个免费开源的平台虚拟化解决方案，允许对一个完整的个人计算机环境进行虚拟化，包括处理器、磁盘、图形适配器和网络设备，支持实现同

一平台同时执行多种操作系统的目的。客户操作系统所产生的任何 I/O 请求都会被中途截获，并重新发送到QEMU进程模拟的用户模式中进行响应。

除了支持CPU、内存和I/O的完全虚拟化，基于KVM的虚拟化技术还提供安全的资源隔离和访问控制、强大的内存管理、基于Linux的硬件支持、虚拟镜像的有效存储、虚拟机动态迁移、设备驱动以及虚拟机可伸缩性和性能管理等功能。

11.4.3 展望

KVM 是解决虚拟化问题的一个创新解决方案，且是第一个进入Linux内核的虚拟化解决方案，可以将标准内核转换成一个系统管理程序，利用内核的优化和改进不断自我发展和完善，因此它的优势不言而喻。从已有的简单CPU性能测评来看，KVM的效率与本地CPU的效率相当，有相当的竞争力，它的出现可能会完全改变使用集群和部署系统的方式。目前为止，虚拟化技术的发展时间不长，KVM 相对来说更是一种比较新的技术，但由于其小巧并效率高的特性，作为Linux内核的有机组成部分，备受瞩目，脱颖而出，越来越受到广大用户的青睐。

虽然KVM发展优势明显，目前也存在两大问题：一是它需要有虚拟化能力的处理器支持，如Intel VT 或 AMD-V 处理器的支持；二是它需要一个用户空间QEMU过程来提供I/O的虚拟化。瑕不掩瑜，作为构建在Linux内核中的KVM，它是对现有虚拟化技术的巨大提升，已经得到了广大用户的青睐，其将来的进一步发展，指日可待。

另外，开源虚拟机（Open Source VM）的兴起，意味着开源架构内涵的进一步丰富和完善，而KVM的创新，则完全遵守开源软件所遵守的GPL（General Public License）规则。随着KVM技术的日益成熟，开源架构的应用必然普及，将推动瘦型计算机产业的兴起和普及。

11.5 Microsoft

11.5.1 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为代表的桌面虚拟化，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。微软公司整合化了虚拟化产品线，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11.11所示。

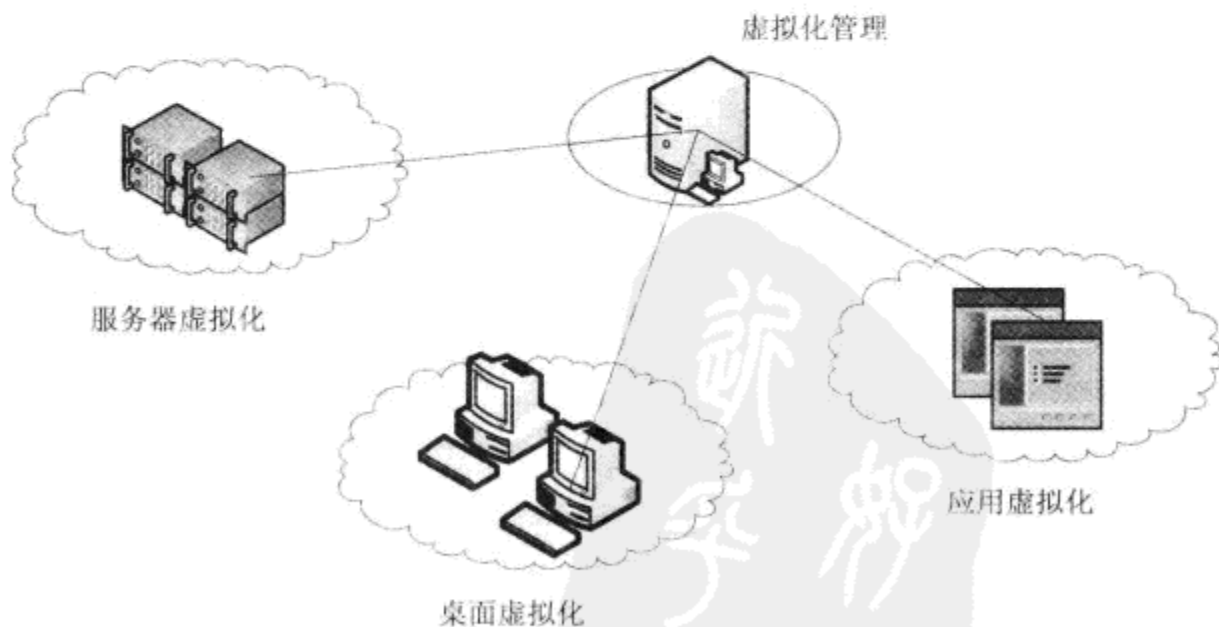


图11.11 微软公司的虚拟化产品

虽然微软公司进入虚拟化市场的时间并不长，但已引起了极大的反响，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。

在接下来的几节里，我们将介绍微软公司的虚拟化产品和相关技术。

11.5.2 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。

11.5.3 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。

11.5.4 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。

11.5.5 虚拟化管理

如果仅有前几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 Manager。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。

11.6 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、面向桌面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。

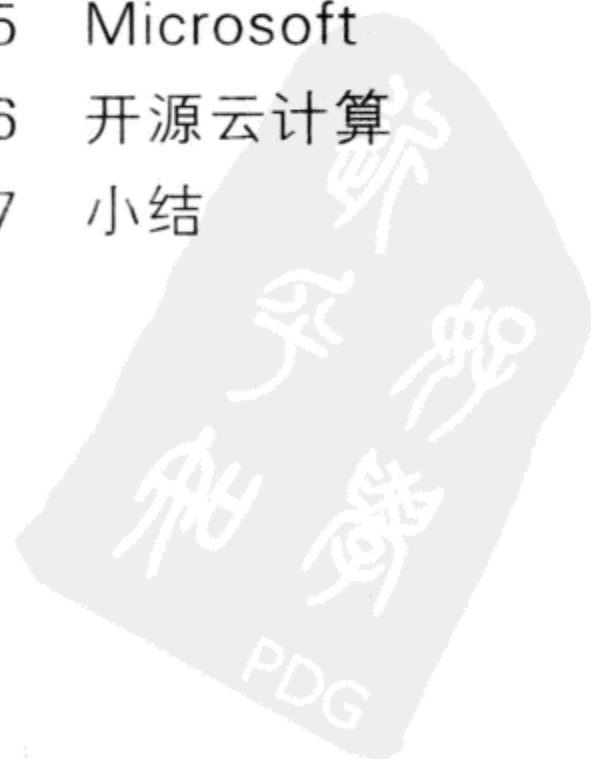
微软公司凭借强大的软件研发实力，在x86虚拟化市场获得了一席之地，产品线也较为全面。我们介绍了微软服务器虚拟化、应用虚拟化、桌面虚拟化和虚拟化管理四个方面的产品，并且解析了微软公司“从数据中心到桌面虚拟化”的战略构想。



第 12 章

云计算的业界动态

- 12.1 IBM
- 12.2 Amazon
- 12.3 Google
- 12.4 Salesforce.com
- 12.5 Microsoft
- 12.6 开源云计算
- 12.7 小结



自2007年起，各行各业的业务人员和技术人员对云计算的关注持续升温。云计算为人们展示了未来信息技术的发展趋势，也为业界带来了巨大的潜在商业价值。根据市场研究机构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，以及它们的产品，帮助读者加深对它们的了解。

12.1 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，也是云计算的先行者和推动者。IBM认为云计算能够提供一种经济有效的业务交付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，提高了信息基础设施对业务需求的响应速度。这种下一代“服务”的交付模式代表了IT发展的最新态势，符合IBM从1995年开始向服务转型的整体战略目标。更重要的是，IBM认为“云计算”理念的生命周期超过10年，符合IBM长期演进的战略构想。因此，IBM以前所未有的力度在推进云计算理念的完善和相关硬件产品、软件产品及服务的研发、生产和推广。

目前，IBM的云计算战略已经形成，它继承并发扬了IBM按需应变（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。凭借在虚拟化、标准化和自动化方面积累的经验 and 雄厚的技术实力，IBM围绕该战略推出了一系列解决方案，为不同的客户量身打造适合他们的云环境。到目前为止，IBM已经帮助分布在全球几十个国家和地区的大量企业和机构搭建了云环境，也在自有的数据中心里搭建开发测试云等多种云环境，以“服务”的形式销售公有云。不仅如此，为了培养云计算的生态系统，IBM在业界推广普及云计算的相关技术，积极组织和参与制定云计算的相关标准，帮助更多的企业和个人了解、使用和研究云计算。IBM还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。

12.1.1 概述

IBM的全方位云计算解决方案以“智慧商务”（Smart Business）为品牌，主要包括以下两个部分。

1. 解决方案

Smart Business系列解决方案包括咨询、设计、实施、运维等端到端服务，根据不同行业、不同场景分为开发测试云、桌面云、存储云、分析云和云咨询等。这些解决方案侧重于帮助客户搭建公有云或私有云环境。

其中，开发测试云（IBM Smart Business Development and Test Cloud）是针对用户开发测试系统而搭建的云环境，用户能够方便地借助云平台实现开发测试资源的申请、监控、回收、盘点等一系列操作，从而提升开发测试的效率和开发测试资源自身管理的效率。桌面云（IBM Smart Business Desktop Cloud）针对客户的PC、PDA、智能手机等多种设备进行分布式部署和集中化管理，从而有效提升终端的安全性和低成本管理的效率。存储云（IBM Smart Business Storage Cloud）主要针对客户存储设备，提供跨设备边界、跨技术形态（如SAN、NAS、Tape等）的统一整合存储平台，提高存储性能和数据管理效率。分析云（IBM Smart Analytics Cloud）是满足客户的海量数据分析需求的信息服务。以商业智能和分析为例，IBM提供硬件、软件 and 相应的服务，在客户的环境中构建私有云。IBM分析云通过集中化和标准化的方式交付商业智能，帮助客户降低成本。

云咨询分为业务咨询服务和技术咨询服务，是IBM针对客户的业务系统和IT系统向云计算演进时的不同需求分别提供的咨询服务，主要帮助客户进行清晰的云演进路线图设计、云启动点分析、云演进收益分析及风险控制等。

2. 软件产品和硬件产品

软件产品包括协作SaaS应用软件LotusLive、PaaS中间件弹性扩展平台软件WVE（WebSphere Virtual Enterprise）、IaaS基础架构虚拟化软件PowerVM，以及云计算管理平台软件TSAM（Tivoli Service Automation Manager）等一系列涵盖SaaS/PaaS/IaaS的端到端软件产品。硬件产品包括整合存储云平台XIV、SAN存储虚拟化管理平台SVC（SAN Volume Controller）等。另外，IBM还推出了IBM CloudBurst，开箱即用的整合云硬件和软件平台以缩短客户在云项目上的开发、部署时间。

对于SaaS应用软件LotusLive，IBM提供了LotusLive、LotusLive iNotes和 LotusLive Integrated Apps三种服务。LotusLive允许客户通过浏览器使用文件共享、网络会议及即时通信等功能。LotusLive iNotes提供了电子邮件、日程安排和联系人管理等功能。LotusLive Integrated Apps服务能够帮助客户开发和LotusLive整合的应用，实现更加简单高效的办公协作。

IBM CloudBurst是一个集硬件、中间件和应用于一身的云环境管理系统。整个系统构建在IBM System x刀片服务器平台上，其管理功能由TSAM（Tivoli Service Automation Manager）软件实现。与客户的IT设施进行整合以后，通过IBM CloudBurst提供的集中自助页面，客户可以设计、部署、管理和监控云环境。

12.1.2 IBM云业务咨询服务

由于云计算能够帮助客户拓展新的业务空间、节约成本及提高业务的灵活性，因此获得了很多客户的青睐。不过，云计算在企业里的实施并不是一个单纯的技术问题，它涉及企业的方方面面，比如企业战略、操作模式、技术集成及可能的组织结构调整。目前，市面上出现了很多云计算相关的产品和服务，怎样选择合适的产品或服务成为企业中战略决策者必须要考虑的问题。

IBM全球业务咨询服务部门能够为客户提供全方面、多层次的云计算咨询服务。具体而言，IBM将这些咨询服务分为战略咨询、应用开发服务和测试服务三个大块。在云计算战略咨询方面，为了更好地帮助客户解决实际问题，IBM将这一块细分为面向云计算提供者的战略及变更服务，以及云环境中的应用安全服务。

先来介绍面向云计算使用者的战略及变更服务。由于云计算必定会给整个社会带来深刻的变革，改变IT服务的交付模式和人们的消费模式，因此云计算提供者必须审慎思考如下几个问题：

- （1）如何能够充分利用这次机遇，从而获得新的客户，并占领新的市场；
- （2）要规划什么样的战略和路线图达到此目的；
- （3）制定好的战略会对企业的组织结构和基础架构有什么影响；
- （4）如何实现按照路线图顺利演进，从而成为成功的云计算提供者。

通过分析客户在云计算方面的机遇和所具备的能力，IBM能够帮助客户解决上面的这几个问题，并制定出相应的战略和路线图。具体而言，可以分为以下三个阶段：分析和设想阶段；战略设计阶段；路线图实现阶段。

在第一阶段，IBM会评估客户的企业战略、云计算的机遇和成熟度，以帮助客户决定如何为新的客户交付云计算服务。第二阶段会制定出云计算的战略和相关的操作模式，这

里需要考虑目标市场的分类、客户成熟度和符合公司目标，同时还会创建一个技术集成报告帮助客户理解交付云服务的架构和模型。第三个阶段会实现路线图，IBM会和客户一起制定出整个流程和相关的项目计划。

对于那些想使用云计算而又不知如何下手的企业决策者们来说，IBM将提供的面向云计算使用者的战略及变更服务无疑是雪中送炭。这些企业决策者们确实认识到云计算改变的不仅仅是企业内部的信息技术基础架构，而是一次从上到下的深刻变革。通过这次变革，企业将会降低对业务和IT支撑系统的运维成本，提高业务和IT支撑系统的效率。IBM能够根据客户的实际情况制定出对应的战略和路线图，并且辅助客户开始这场变革。

第一步，IBM利用一套经过验证的方法论和评估工具开发一套适合客户的云计算战略，在这一步中，IBM会分析和评估客户的人员、流程、组织结构、管理模式、基础设施、技术等方方面面。比如，如果客户需要业务流程即服务，那么计费、合同管理、人事管理等流程就需要详细分析；如果客户需要基础设施即服务，那么就需要分析和评估客户的服务器、存储及计算能力这些基础设施资源。

第二步，IBM会帮助客户细化定制化的云计算战略，并建立相应的云计算模型，从业务层面来讲，包括了详细分析云计算实施的商业价值、制定相关服务的优先级、分析投资回报率；从技术层面来讲，包括了IT战略、技术架构和实现的相关路线图。在最后的第三步，IBM会和客户沟通以后实施一个局部的示范项目。这是至关重要的一步，因为这样可以验证制定的计划是否能够达到预期目的。

云环境中的应用安全服务是解决云计算中潜在的应用安全性和数据隐私性问题。毫无疑问，云计算给人们带来了诸多好处和便利，但是我们也常听到人们对云计算安全性问题的担忧。IBM通过对客户的大量调研，总结了人们对于云计算安全性的几点担忧：

(1) 失去控制权，采用云计算以后，客户可能不再拥有自己的基础架构，而是使用云计算提供商的；

(2) 兼容性风险，由于数据和系统不受客户直接控制，云计算提供商在这方面可能采用与客户不同的规章和标准；

(3) 隔离失效，多租户技术和资源共享是云计算的特色，但是如果两个竞争企业使用了相同的云服务，该服务能否保证同等处理它们的工作负载，并且做到隔离它们的内存、存储和网络访问；

- (4) 数据保护，采用云计算以后，客户的数据保护完全依赖于提供商；
- (5) 管理接口和基于角色的访问，云计算能否保证信息的访问控制。

为了分析、量化和解决客户的这些顾虑，IBM提供了相应的安全评估，该评估从客户自身和客户所采用的云计算服务两方面入手，涵盖身份和访问管理、数据保护、审计和监控，法律、规章和隐私需求等内容，帮助客户找到一个既能充分利用云计算的优势，又能最大限度保留控制力的平衡点。

除此之外，IBM还提供了基于云计算的应用开发和测试服务。应用开发服务是帮助客户制定开发云应用的技术路线图，并且从流程、设计、解决方案架构、系统集成、安全和变更管理方面指导客户，从而使得客户能够最大限度地挖掘云应用的商业价值。测试服务不仅能够帮助客户减少在测试阶段的成本开支，还能加速业务的快速上线。

12.1.3 IBM云计算基础架构策略和计划服务

对于一个企业来说，采用云计算对它的业务和IT系统都有影响。12.1.2节提到的IBM云业务咨询服务更多地是从企业的业务角度解决这个问题，而这一节所提到的IBM云计算基础架构策略和计划服务更多地是从IT系统角度解决问题。

IBM云计算基础架构策略和计划服务主要帮助客户解决有落地意义的云演进路线图、有效的IT架构设计和风险可控的云项目切入点三个问题。

具体而言，IBM云技术咨询服务分成五个步骤实现，如图12.1所示。

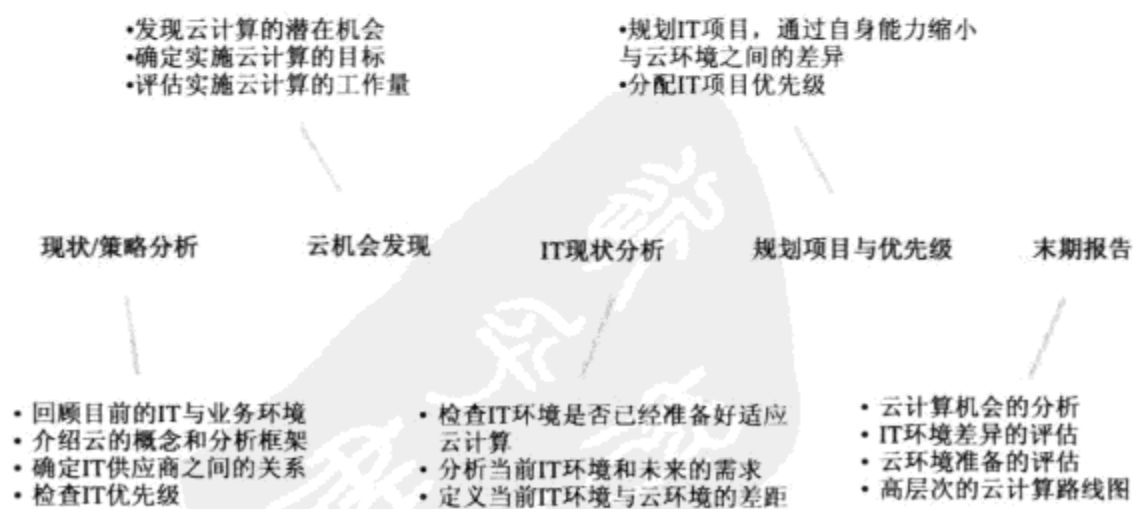


图12.1 IBM云技术咨询服务的五个实现步骤

1. 现状与策略分析

访谈、回顾目前的IT和业务的环境，引入云计算理念和分析框架，决定IT提供者关系和影响，并回顾IT的优先级。IBM将基于如图12.2所示的IBM云采纳分析框架进行分析和评估。同时，IBM还将基于IBM CUPE (Commodity,Utility,Partner,Enabler) 分析框架进行目前与将来的IT和业务的关系分析。



图12.2 IBM云采纳分析框架

2. 云机会发现

发现潜在的云机会领域、决定预期云计算目标，评估潜在的云负载情况。在评估负载时，IBM将基于如图12.3所示的负载分析工具进行分析。

Workloads	HW, SW, Facilities Maturity	Virtualization Maturity	Utilization Improvement Opportunities	IT Automation Maturity	Workload Standardization Opportunities	Data Constraints	Business and IT process improvement opportunities	% of Total Revenue Affected	% of Total Cost Affected
1 Web Servings	Low	Low	High	Med	Med	Med	Low	11-20%	11-20%
2 Web Applications	High	Med	Med	Med	Low	Low	High	11-20%	21-30%
3 BI Data Warehouse	Low	High	Low	High	Low	High	Low	11-20%	11-20%
4 ERP, SCM	High	Low	Med	High	Low	High	Med	0-10%	31-40%
5 Analytics	Med	Med	Low	Med	Low	Med	Low	11-20%	11-20%
6 Numerical, Batch	Med	Med	Med	Med	Med	Med	Med	0-10%	0-10%
7 Collaboration	Med	Low	High	Low	Med	Low	Med	0-10%	9-10%
8 File & Print	Low	Low	High	Low	High	Low	High	0-10%	0-10%
9 Desktop	Med	Low	High	Low	Med	Low	High	0-10%	21-30%
10 Development & Test	Med	Low	Med	Med	High	Low	High	11-20%	11-20%

图12.3 IBM的负载分析工具

3. IT现状评估

回顾往云迁移的IT前置条件，分析当前IT状况和未来在云环境下的IT需求，定义其间

的差距。图12.4展示了这种差距分析的示例。

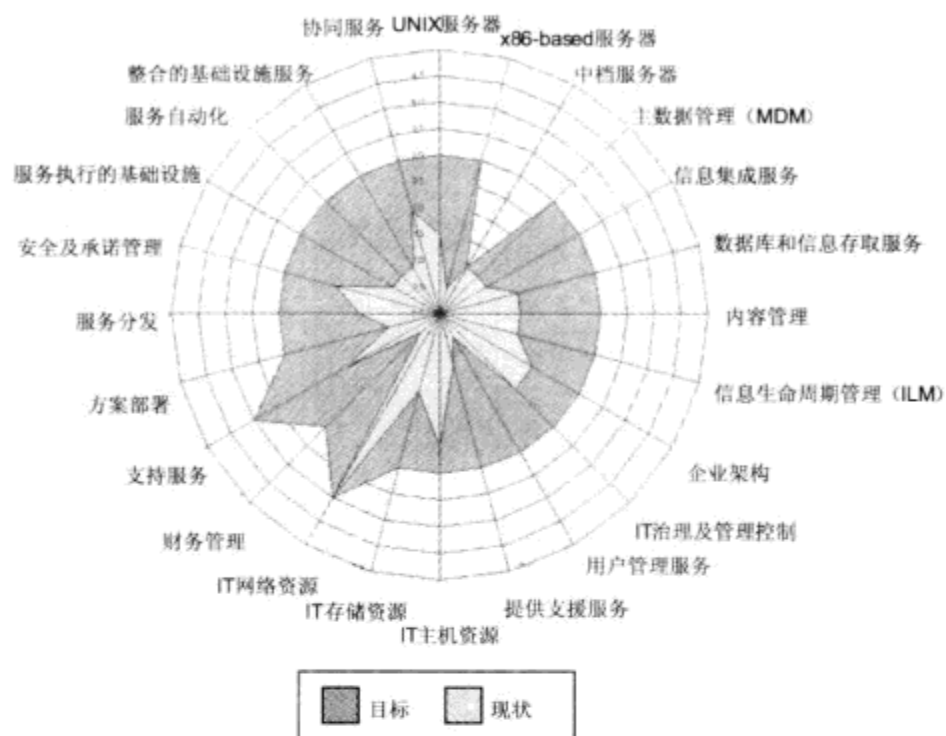


图12.4 差距分析示例

4. 规划项目与优先级

根据IT现状与云环境IT需求差距分析的结果，规划出若干IT项目，通过这些项目缩短与云环境需求的差异。对于每个IT项目，应综合考虑私有云、公有云和混合云的特点，采用相应的模式，并根据实际能力和需求重点，确定这些IT项目的优先级及投资花费。

5. 末期报告

末期报告主要包括如下内容：首先是概述部分，概括描述整个项目。随后是背景介绍，比较详细的介绍项目和客户的背景。第三部分是客户的状况，主要包括客户的主要业务驱动、客户的主要IT项目、客户的主要IT方向和优先级。第四部分是客户的云的策略性方向，会包括概述、云的模式、云模式中的角色、最适合云模式的资源、最适合云模式的工作负荷。在确定了方向之后，报告会分析客户当前状况与期望方向的差距，即差距分析。随后的部分是制定客户发展的路线图，再给出IBM的观察、理解和建议。最后给出结论，并指明下一步要做的事项。在报告最后的附录部分，可以附上详细的路线图，以及对于评估结果的详细描述。

12.1.4 IBM Smart Business Development and Test Cloud

如今，以下问题通常困扰着身处不同行业，拥有软件开发测试部门的客户。

1. IT资源管理不灵活

开发测试工作需要使用的IT资源有服务器、存储、操作系统、中间软件等。这些资源通常由开发组和测试组申请和使用，由基础架构组负责运维和管理。中大型客户通常同时进行多个项目的开发和测试，长期下来，究竟总共有多少资源，其中哪些被使用，哪些被空闲，哪些将被归还等涉及资产管理的一系列问题通常会难以解答。这严重影响了开发测试的效率提升。

2. 开发测试资源申请周期长

有数据显示，一个为期6个月的开发测试项目会有1个月时间花费在申请资源、审批资源、等待资源、安装资源、配置存储、按照开发测试需要配置资源、拆除资源、归还资源等重复性工作上。这些重复性工作，严重拖延了开发测试工作的效率。

3. 员工工作效率不高

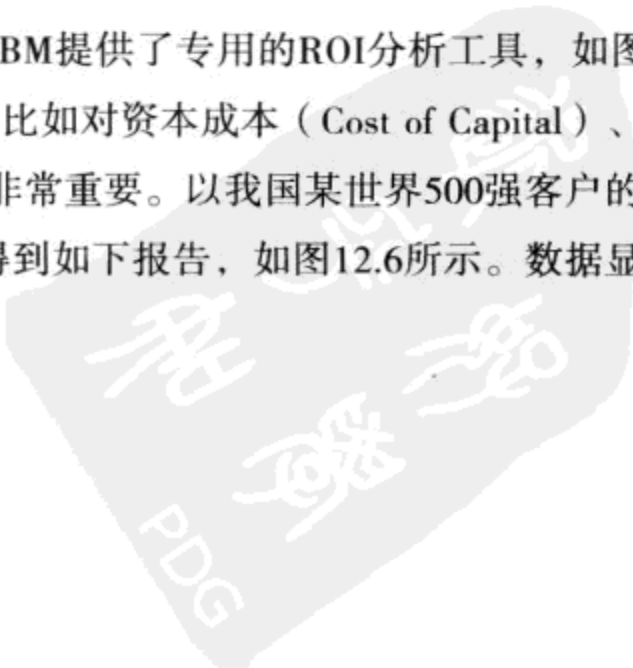
除了以上重复性工作，负责进行资源维护的基础架构工作组通常需要频繁地制作软件镜像文件、更新服务器等硬件设备的固件等。这在很大程度影响了员工技能的提升。实际上，很多基础架构工作组更愿意做如错误诊断、性能调优等高附加值的工作。员工工作效率不高，也影响了开发测试部门的整体管理效率。

4. 设备利用率不高

有大量的实际调研数据显示，开发测试所用服务器的平均CPU利用率不到5%，以太网带宽利用率不到10%。这些资源浪费的现象阻碍了客户投资回报率（Return On Investment, ROI）的提升。

在这种资源管理不灵活、效率低、重复性高的背景下，开发测试云的理念应运而生。实际上，Google当初提出云理念，也是由开发测试的具体环境启发的，而目前在业界应用最广泛的云案例，也大多与开发测试相关。

为量化客户在开发测试云的投资回报，IBM提供了专用的ROI分析工具，如图12.5所示。在进行ROI分析时，分析师的财务背景，比如对资本成本（Cost of Capital）、资金净现值（Net Present Value）等财务术语的理解非常重要。以我国某世界500强客户的实例为例，在工具相应的表格输入内容后，我们将得到如下报告，如图12.6所示。数据显示，此实际客户的每年投资回报率在50%以上。



Item	Default value	Client Provided (overwrite default value)	Considered for Analysis
Hardware			
Number of existing physical servers	Need Data		Need Data
Average total purchase price per server	Need Data		Need Data
Average Hardware % utilization	20%		20%
Services Charges (Only for outsourced accounts)			
Monthly services charge per physical server per month	\$0		\$0
Average monthly per-image instance managing cost	\$0		\$0
Image Assumptions			
Number of image instances provisioned per year	Need Data		Need Data
Average number of image instances being actively managed	Need Data		Need Data
Number of existing images to be migrated or recreated to cloud environment	0		0
In average, how many years is a image instance actively managed	0.5		0.5
How many system administrators are managing this environment? (Network and storage administrators are excluded)	#VALUE!		#VALUE!
Software Assumptions			
For SO accounts, who pays for OS and VMWare licenses?	Customer		Customer
Number of existing non-Free OS (e.g. Windows Server) licenses	Need Data		Need Data
Cost of non-free OS license (one time cost) per physical server	\$1,500		\$1,500
Cost of non-free OS annual license maintenance per physical server	\$300		\$300
Labor Assumptions			
Gross annual cost of labor (admins, testers, etc)	\$120,000		\$120,000
Total work hours per year	1920		1920
Who does OS-level admin?	Customer		Customer
Provisioning Process			
Average FTE hours required per image provisioning for customer	40.00		40.00
Average FTE hours required per image provisioning for IBM	0.00		0.00
Testing Process			
Total number of projects per year	Need Data		Need Data
Average idle time caused by provisioning per project	20		20
Misc			
Cost of capital	20%		20%

图12.5 IBM专用的开发测试云ROI分析工具

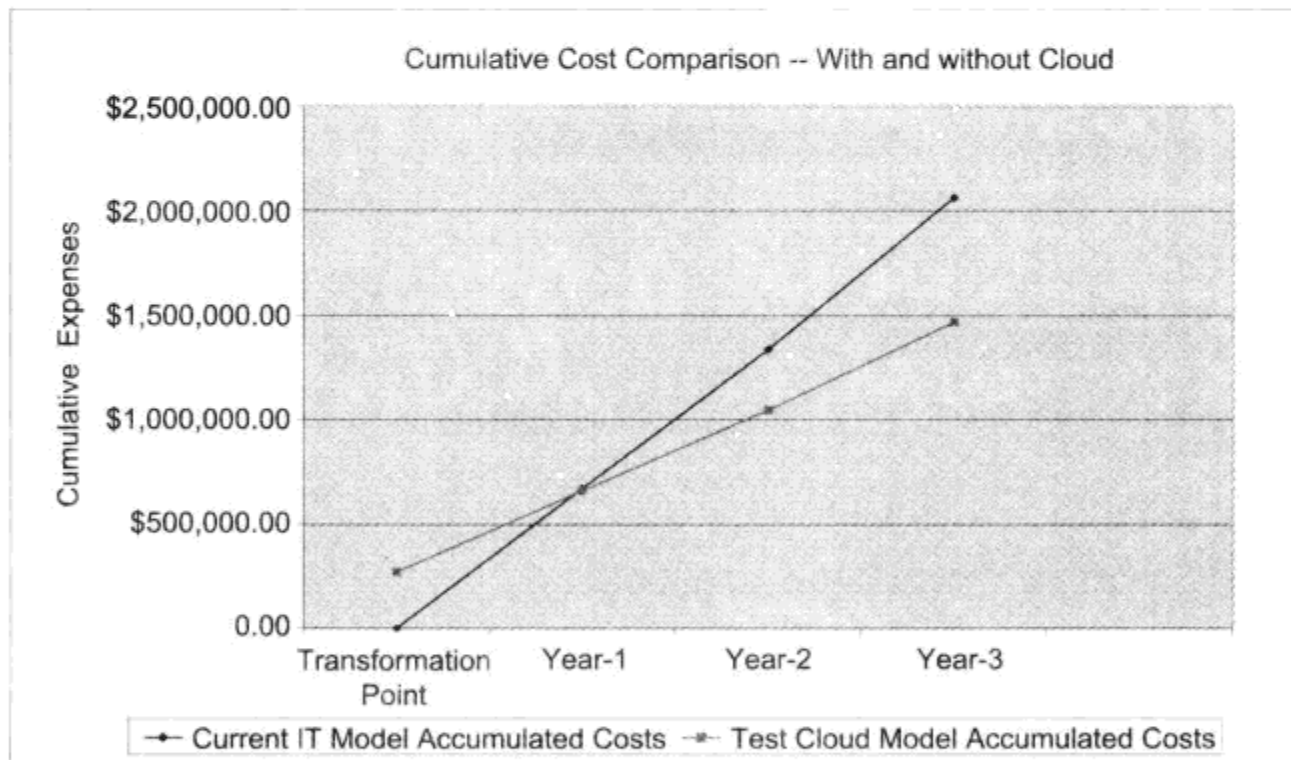


图12.6 IBM专用的开发测试云ROI分析结果

IBM的开发测试云解决方案主要包括如下内容。

1. 标准化资源池的建设

建设标准化的软件、硬件资源池。进行标准化资源池建设需要虚拟化技术的支持，比

如利用PowerVM进行IBM小型机的虚拟化建设，利用Xen进行x86平台的虚拟化建设，利用SoNAS进行存储的虚拟化建设等。同时，IBM也会根据OVF等标准化组织的标准和特定企业自行定义的标准，进行软件的标准化建设。

2. 云管理平台的建设

主要包括三个方面，一是服务目录建设，二是资源管理流程建设，三是资源调度集成。所谓服务目录，是指从单元测试开始到用户接受度测试的整个开发测试过程对基础资源池的要求，以服务形式（也就是开发测试团队可申请的资源包）体现的资源目录。比如，一个典型“服务”是：2C@IBM p/ 64G Mem/ 512G SAN Storage的服务器存储平台，上面预装了AIX6.1/WAS7.0/DB2 9.5。这个服务一般在单元测试阶段申请。所谓资源管理流程，是指基于ITIL的运维体系，进行资源申请、审批、使用、提示、回收、盘点等资源生命周期各阶段所涉及的组织、人员、权限、安全、计费、审计等一系列管理流程的设计和实现。所谓资源调度，是指标准化资源池底层管理软件（如TPM）与云管理平台软件（如TSAM）的集成。

3. 系统集成

主要与身份认证系统（如IBM Tivoli Directory Server或LDAP Server等）、监控及流程系统（如IBM Tivoli Monitor）、计费系统等系统集成。

4. 系统迁移

也就是将现有的开发测试项目迁移到开发测试云平台上。如何做到在保证数据不丢失的前提下不影响开发测试组的进度，是这个过程的关键问题。

12.1.5 IBM Smart Business Desktop Cloud

回顾计算机的发展历程，人们能够直接接触到的设备——鼠标、键盘没有任何变化，显示屏略微改变，而主机部分，也就是CPU、内存、显卡、主板等则有翻天覆地的变化。实际上，企业历代更新电脑时，采购的是包括外设和主机在内的全新电脑，这就产生了较大的浪费。

基于这个发现，一个创新的想法产生了：能否把不变的部分依旧放在员工的办公桌上，把变化的部分——主机集中起来放在数据中心里？这种想法，正是桌面云服务方案能够产生和流行的最主要源动力。

IBM为某大型客户实施桌面云方案时实际的ROI分析结果如图12.7所示。这个实例中的客户采取单网络方案。所谓单网络，是指每个桌面（员工）只配置一台电脑。在这里，逐年累积上升的ROI投资回报主要来源于如下几个方面。

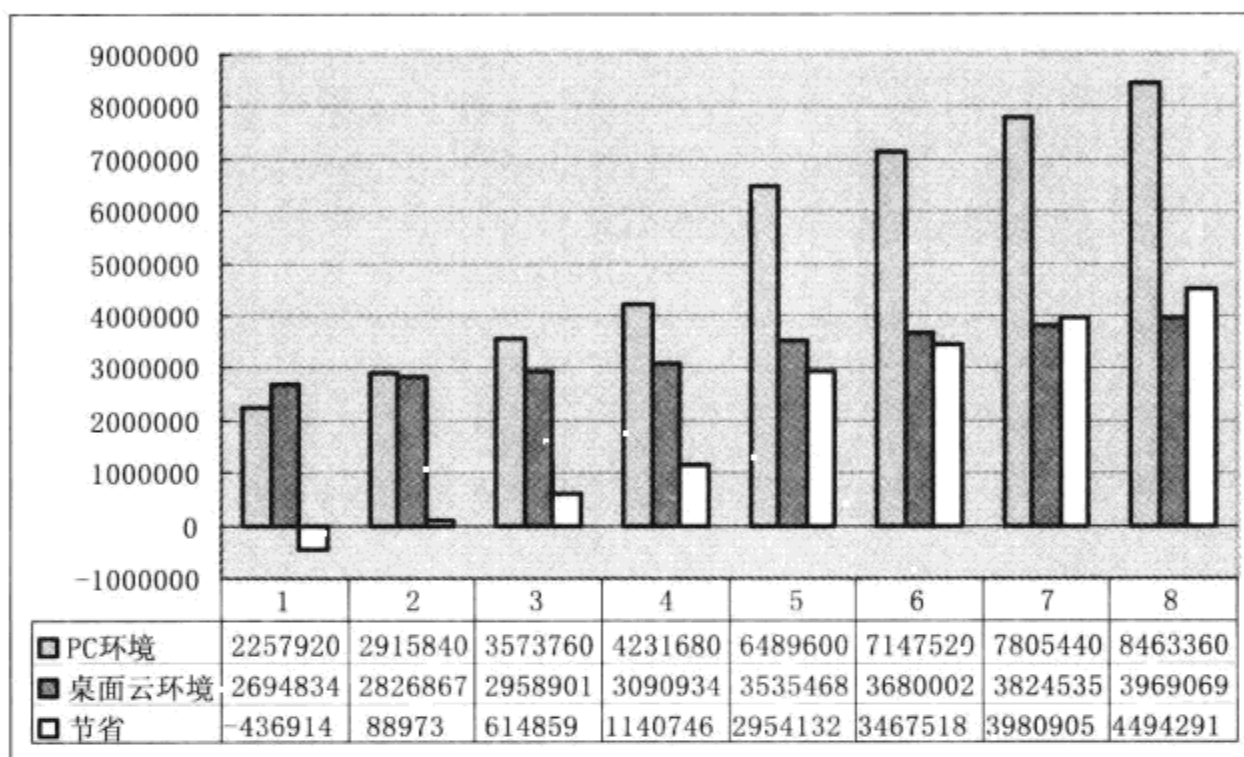


图12.7 IBM为某大型客户实施单网络桌面云方案时的ROI分析结果

1. 采购成本

在实施桌面云方案前，每台电脑按6000元人民币计算，在采用桌面云后，每个桌面设备不到2500元人民币，而主机设备，只需配置3台4核x3850服务器。若客户总共有400台电脑，这就直接节省了上百万元的硬件设备成本。此外，客户原先还需采购终端管理软件进行自动软件更新、桌面远程维护等操作，以5年计平均每台需4000人民币。桌面云能够直接在数据中心端维护，也直接节省了上百万元的终端管理软件成本。

2. 使用成本

根据业界的统计数据，平均每200台电脑需要1个专门的维护人员，这就产生了大量的维护成本。此外，从电费角度，由于没有终端主机设备，桌面云方案的电量不到原先方案的20%。考虑到每台电脑约200瓦的功率，节约的电费也将是可观的。

如果我们考虑一机双网（在政府行业中这种应用场景非常普遍）的场景，以图12.8的ROI分析将更加突出。

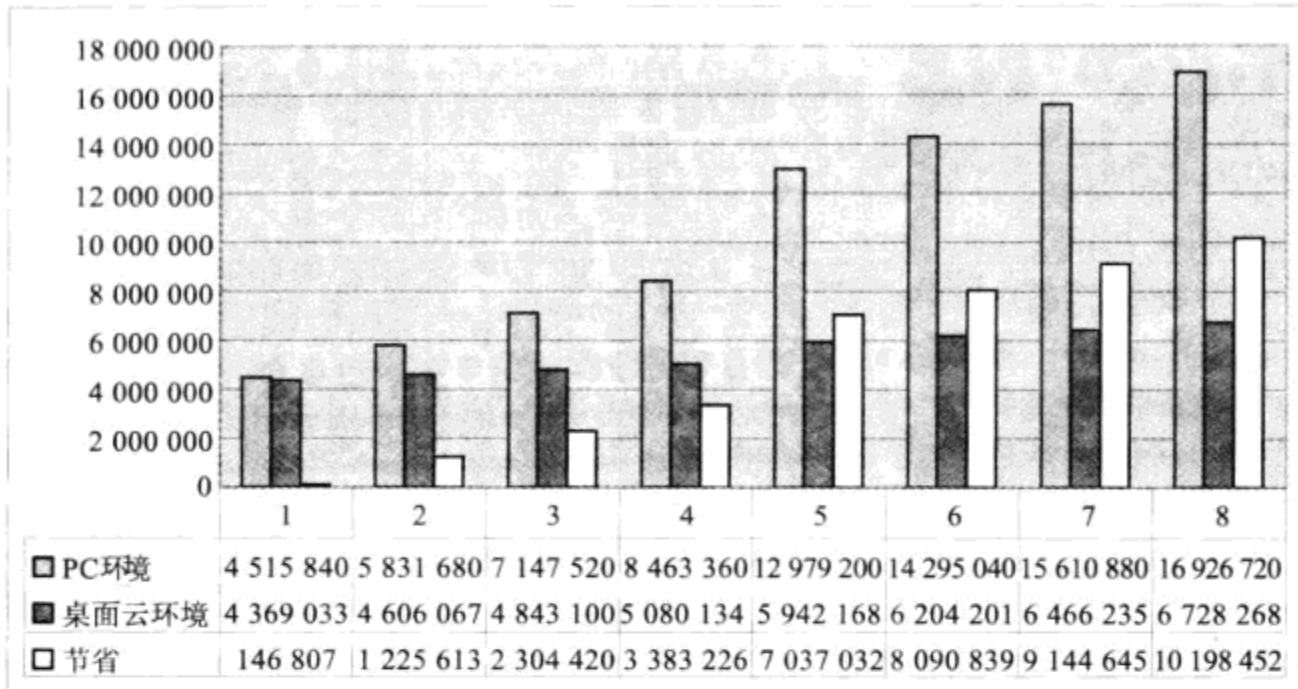


图12.8 IBM为某大型客户实施双网络桌面云方案时的ROI分析结果

此外，在办公室里，电脑主机的噪音和辐射对员工的工作舒适程度是个不小的挑战，而在桌面云解决方案中，前端不会有任何主机设备产生噪音。在桌面云的解决方案中，由于数据存储和运算都将在服务器端进行，客户端只显示镜像内容，因此，桌面云在安全控制上也有非常突出的效果，同时应用程序无须做任何修改，部署风险小。综合考虑这些因素，不难发现桌面云解决方案的优势非常明显。

因此，很多客户愿意优先部署桌面云解决方案。而IBM也根据不同的应用场景，提供了四种富有针对性的桌面云解决方案，如图12.9所示。



图12.9 IBM提供的桌面云解决方案

不论何种场景，IBM的解决方案都具有如下的特色功能：低成本低复杂性的桌面虚拟化、高安全性的应用和数据管理，以及灵活的资源使用模式。

低成本低复杂性的桌面虚拟化主要来源于IBM在分布式桌面管理领域（DMS, Desktop Management Service）和管理公司自身超过40万台员工电脑时积累的丰富经验。这些经验迁移到集中式桌面管理，也就是桌面云管理上同样具有巨大的价值。比如，给Windows XP操作系统打SP2补丁。由于SP2默认打开防火墙，这与客户自行部署的防火墙有一定的冲突。如何有效避免这些冲突，这是不论在分布式桌面管理还是桌面云管理时都会面临的问题。

IBM正是借助这些经验，在桌面云领域获得了客户的认可。

高安全性的应用和数据管理主要体现在三个方面：服务器端的数据存储和运算、客户端的数据备份和恢复，以及统一的客户端安全控制策略。在IBM的桌面云方案中，数据存储和运算都将在服务器端进行，客户端只显示变化的镜像。客户端操作产生的数据将保存在服务器端，服务器端采用虚拟桌面基础设施和一系列的安全控制手段来保证数据和应用的安全性，就像将钱存入银行比存在家里更安全一样。将敏感数据保存在数据中心，由于数据除了在终端屏幕上显示外，并不会保存在终端，这大大减轻了终端数据保护的壓力。另一方面，只显示变化镜像的客户端难以从服务器端窃取未经授权的资料，采用虚拟桌面的客户端的行为和状态也能够被数据中心审计，从而评价客户端的安全控制手段是否符合法规，对可能的恶意攻击增强防范措施。IBM的桌面云方案还可以动态地在数据中心根据相应的性能和可用性对远程客户端提供虚拟桌面的数据备份和恢复。此外，它还采用集中式的IT环境管理和加密的传输，对关键性应用和数据的访问进行了控制，根据安全挑战制定出各种各样的安全规则。

这些规则可以迅速作用于每个客户端桌面，实现对客户端的严格保护。

灵活的资源使用模式主要体现在灵活的服务交付方式、定价策略和对公有云和私有云模式的支持。无论客户使用的是私有云还是公有云，选择何种服务交付方式，采取何种定价策略，IBM桌面云解决方案都能提供对应的资源使用模式，帮助客户剔除掉原先使用和管理物理资源所带来的高成本和高风险，真正获得通过虚拟桌面进行资源利用的好处。

IBM桌面云解决方案提供的上述三个主要功能，是基于如下基础架构实现的。

► 瘦终端，可以通过各种协议连接到运行在服务器上的桌面设备。为了充分利用已有

资源，实现IT资产的最大化应用，该架构也支持对传统桌面做一些改造，通过安装插件，使得它们也有能力连接到运行在服务器上的桌面。

- ▶ 网络的接入，提供了各种接入方式供用户选择，比如有线或者无线网络连接。连接的网络可以是局域网，也可以是广域网。连接的方式支持正常连接和安全连接。
- ▶ 控制台，对运行虚拟桌面的服务器进行配置，如配置网络连接、存储设备等，还可以监控运行时服务器的一些性能指标，如内存的使用情况、CPU的使用率等。
- ▶ 身份认证，提供用户的认证和授权进行安全认证。
- ▶ 操作系统或应用程序，采用共享服务的方式提供桌面和应用，可以在特定的服务器上提供更多的服务。
- ▶ 应用服务器，把各种应用分发到虚拟桌面，客户只需要连到一个桌面就可以使用所有的应用，就好像这些应用安装在桌面上一样，这种架构提供了与使用传统桌面完全一样的用户体验。

12.1.6 IBM Smart Business Analytics Cloud

对于任何企业，尤其是大型企业，信息是至关重要的。在竞争激烈的市场环境下，谁能够及时获得有用的信息，谁就赢得了主动权。当然，信息并不是凭空产生的，而是通过对大量数据进行分析而产生的。这些经过分析和处理的信息能够帮助企业的决策者们做出明智的商业决策，这也是人们常说的商业智能和分析。

为了满足企业客户对商业智能和分析方面的需求，IBM推出了“智慧分析云”（IBM Smart Business Analytics Cloud）解决方案。它是一套包括了硬件、软件和服务的完整解决方案，能够在企业客户的数据中心里创建一个简单、安全、高可用的提供商业智能和分析服务的私有云。不同于其他商业智能和分析的解决方案，IBM的分析云解决方案通过集中化和标准化的方式整合了企业客户各个部门在不同业务上对商业智能和分析的需求，帮助企业客户极大地缩减了这方面的开支。下面，我们来剖析该解决方案的组成部分。

在硬件方面，分析云采用了IBM大型机（System z），它能够满足大型企业对于可靠性、

可用性和安全性的要求，并且在大规模并行处理和z/VM虚拟化技术上也处于业界领先地位。由于这些特性，IBM大型机为上层业务提供了一个高效的集中的硬件平台。

在软件方面，分析云的商业智能和分析功能由Cognos8 BI提供，该软件提供了多种商业智能、分析和性能管理方面的能力，比如通过报表和仪表盘等服务来监控业务的性能、分析趋势和衡量结果。

在服务方面，IBM具有雄厚的技术实力，能够帮助企业客户转变商业智能的战略，并且设计、实现和优化客户的分析云，从而使客户可以快速地获得投资回报。

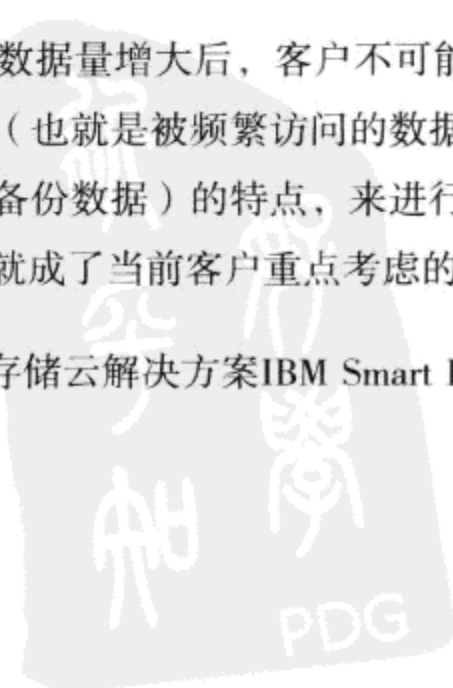
值得一提的是，IBM自己就拥有这样一个分析云——“Blue Insight”。在IBM内部，有超过20万员工正在使用它。通过它的帮助，销售团队能够在全球范围内深度挖掘客户关系，并且获得何种产品或服务能够给客户带来更大的价值的预测信息；产品团队能够迅速地分析销售信息、行业趋势及客户期望，从而来调整产品的功能特性和研发进度。通过采用Blue Insight分析云，IBM预计在未来五年内会减少超过两千万美金的开支。

12.1.7 IBM Smart Business Storage Cloud

十年前采购的电脑硬盘基本在8GB左右；而现在硬盘容量基本在160GB左右。十年前，硬盘转速，也就是数据存储性能最重要的技术指标，在5400转/分钟；而现在，硬盘转速在7200转/分钟。换句话说：存储容量十年间变大了20倍，但存储的性能只加快了不到20%。因此，存储领域基于电子的容量与基于机械的性能这两种技术的不匹配引发了跟存储相关的诸多问题。

同时，由于数据量增大后，客户不可能都采用高端SAN产品来存储全部的数据，如何根据“热数据”（也就是被频繁访问的数据，如生产数据）和“冷数据”（也就是使用频率低的数据，如备份数据）的特点，来进行分级存储，并借助与存储云的解决方案来方便管理这些数据，就成了当前客户重点考虑的问题之一。

IBM提供的存储云解决方案IBM Smart Business Storage Cloud包含两大部分，如图12.10所示。



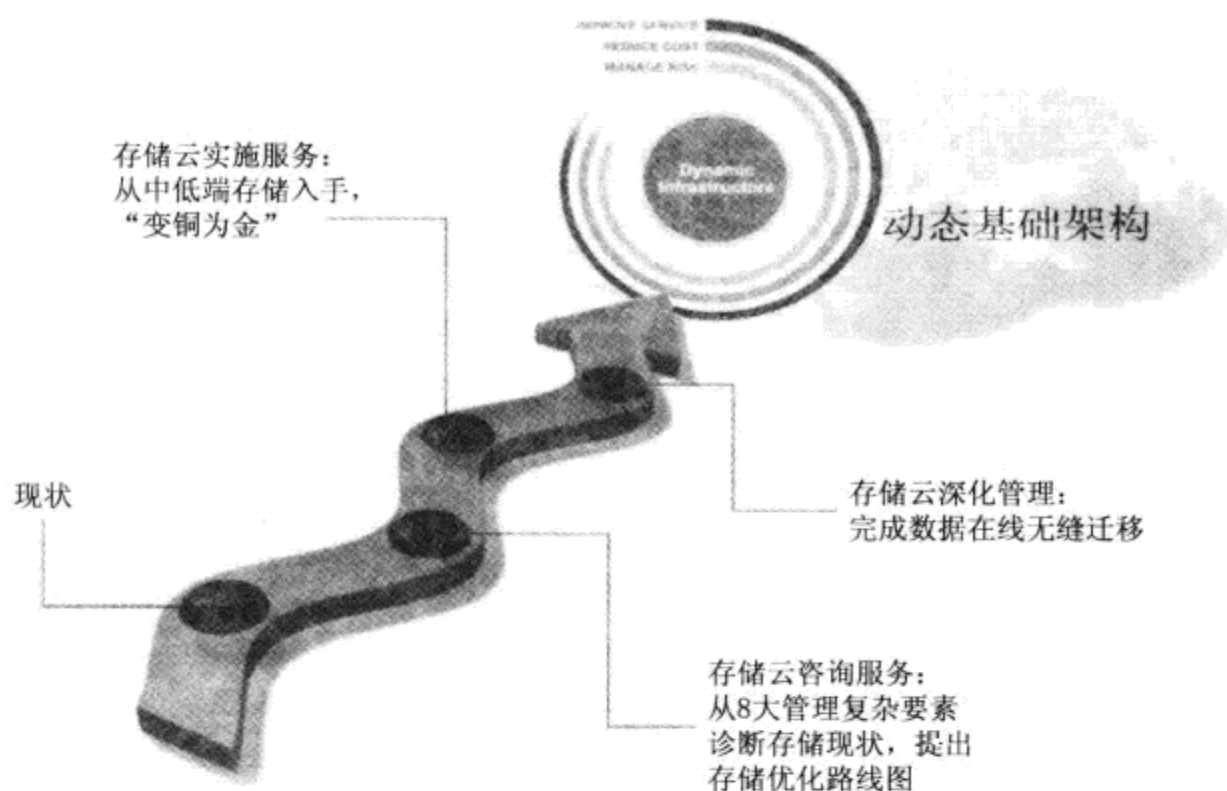


图12.10 IBM提供的存储优化路线图示例

1. 存储云咨询服务

主要通过访谈了解客户的需求，并制订项目计划；确认包括性能、容量、可用性及标识特定待迁移数据在内为客户系统运行现状给出量化评估（其结果可作为行业参照指数），提供一份未来6至18个月的存储优化路线图，指导系统到达最佳运行状态。

2. 存储云实施方案

主要是以Scale-out File Services (SOFS) 技术为核心，帮助客户实现包括动态存储管理、可伸缩功能和性能、并发、多协议数据访问和新级别的可管理能力等在内的存储及存储管理能力。其中，动态存储管理支持多种类型的存储和存储连接机制，使客户能够将光纤通道磁盘、串行SCSI扩展器 (SAS) 和串行高级技术附件 (SATA) 磁盘技术等存储技术相混合从而产生多个存储池，并且还能够依据预定义的政策将数据动态有效地从一个存储池转移到另一个存储池，同时还不会增加额外的开销或者影响数据访问。可伸缩功能和性能是指IBM存储云支持数十亿个文件和PB级的数据，并且动态地提供增长空间以支持未来技术的发展，允许基础设施按需扩展。并发、多协议数据访问支持使用多个标准接口提供对一组公用数据的多协议访问，并且还可以提供增加运行客户自己的应用程序的Linux节点的能力。新级别的可管理能力支持通过一个基于Web的管理工具，提供单一的管理节点管理集群，监管整个解决方案内的事件，支持客户在保持数据对最终用户可用的同时增加或

移除节点和存储，还可以提供新模式的空间管理。

12.1.8 IBM LotusLive

据媒体报道，IBM员工在2009年使用LotusLive进行视频会议和办公协助总计超过10万分钟。IBM LotusLive是IBM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，它具有很多优势。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。

LotusLive提供的办公协作服务有：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及一系列附属插件，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。

12.1.9 IBM TSAM

在现代社会中，各个企业核心业务的发展和信息技术是密不可分的。这些业务不断地要求更强大的计算能力、更快的网络带宽、更灵活的软件架构和更优质的信息服务作为支撑。同样地，信息技术的发展方向越来越受到上层业务的影响，人们希望信息技术不是单独的一堆硬件和软件，而是像服务方便灵活，于是人们对于IT服务的需求催生了面向服务架构（SOA），Web 2.0和云计算等概念。

作为IBM在IT服务领域的重要软件产品之一，IBM Tivoli Service Automation Manager（TSAM）基于虚拟化平台为客户提供了管理IT服务生命周期的方案。这些虚拟化平台包括z/VM、PowerVM、VMware ESX、KVM和Xen。TSAM依据ITIL V3的最佳实践经验来管理IT服务的生命周期，针对IT服务的不同阶段，分配了不同的管理角色，这样既保证了IT服务

管理的一致性，又有针对性。TSAM的设计使得客户能够实现更快速的服务响应和交付能力，以及更低的运营成本。

为了更好地抽象现实中的IT服务，TSAM提出了服务定义（Service Definition）模型。服务定义模型包含两个方面：IT服务的逻辑拓扑结构和流程管理。拓扑结构描述了IT服务由哪些功能组件构成，以及各个功能组件的关系；而流程管理通过管理计划（Management Plan）描述了该IT服务从模板到实例需要经过哪些步骤，某个步骤是否需要经过审批，以及在运行时支持哪些行为。按照ITIL V3的分类，服务定义在服务模板阶段创建，在过渡阶段被实例化，在运行阶段接受动态管理。

一旦服务定义被实例化，TSAM用服务部署实例（Service Deployment Instance）这一概念来描述实例化的服务，它主要包括两部分内容：服务实例的物理拓扑结构和向外提供服务定义模板中的管理计划。物理拓扑结构是指该IT服务的各个功能组件具体所在位置的拓扑结构，比如WebSphere应用服务器集群的管理节点和一个被管节点位于虚拟机A上，另外两个被管节点位于虚拟机B上，它们之间就构成了一个物理拓扑结构。管理计划是在服务模板阶段中规定、在服务运行阶段激活的，这些功能能够动态地改变IT服务的拓扑结构，比如WebSphere应用服务器集群的例子中，可以动态启动或停止一个被管节点。

TSAM端对端的使用场景描述如图12.11所示。从IT服务的设计、过渡和运行，TSAM都提供了完备的、面向服务的功能。

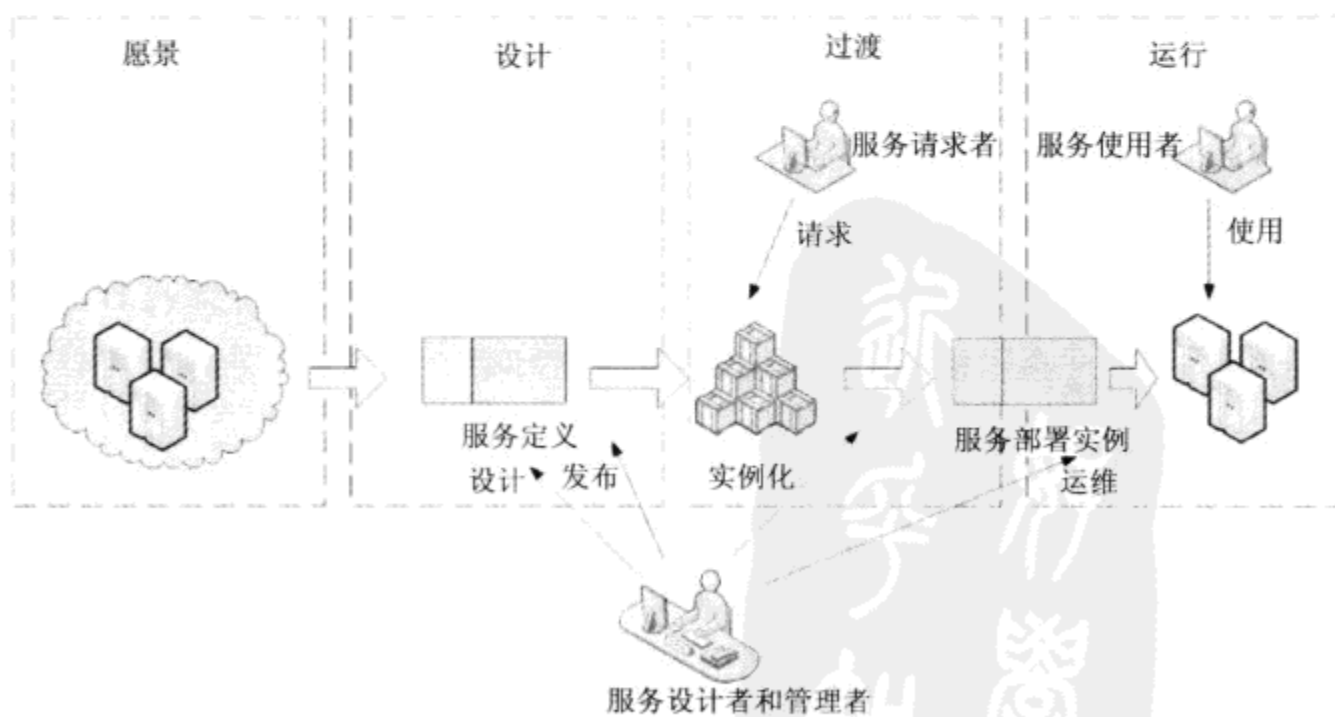


图12.11 IBM TSAM的端对端的使用场景描述

服务设计者根据业务的真实需求，在设计阶段创建了服务定义，这时的服务定义是以模板的形式存在的。服务设计者可以从头开始设计服务定义，也可以借助TSAM提供的丰富的预置服务定义来简化设计工作。

服务定义模板创建完成之后，服务的设计者将它发布到服务定义目录（Service Catalog Offerings）中，提供给服务请求者或最终用户使用。请求者可以在服务定义目录中进行查询，选择自己需要的服务定义，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，为服务请求者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。部署完成以后，服务定义模板被实例化为运行的服务部署实例。

部署完成以后，最终用户可以通过网络方式访问IT服务，服务管理者则在后端负责监控和管理IT服务。TSAM为服务管理者提供了管理计划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应用服务器被管节点时，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。当每一项步骤都被敲定，并确认系统当前可以实现这些操作后，这个WebSphere应用服务器被管节点才能加入到现有服务中。

当IT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，以便日后查询和审计。

值得一提的是，服务设计者、服务管理员所看到的视图和服务请求者、最终用户看到的视图是不同的，TSAM给前者提供的是一个具有丰富功能集的视图，他们可以通过这些功能对服务定义模板和服务部署实例进行管理，而后者的视图是一个简化的自助服务页面，他们只需要进行一些简单的选择，便可以获得所需的IT服务，另外，该视图还可以根据最终用户的需要进行定制。

不仅如此，TSAM还能够和IBM其他IT管理软件进行整合，实现更加丰富的管理功能。例如，和ITM的整合能够实现对IT服务的资源使用状态和工作负载进行有效监控；和TUAM（Tivoli Usage and Accounting Manager）的整合能够实现对IT服务的资源使用情况进行记账和收费；和WCA（WebSphere CloudBurst Appliance）的整合能够实现部署和管理基于

WebSphere的SOA应用服务环境。

综上所述，TSAM作为IBM云计算战略中的重要产品之一，提供了强大的IT服务管理功能，它能够有效地整合客户数据中心的虚拟化资源，提高IT服务响应和交付的能力，为客户降低运维成本。

12.1.10 IBM WebSphere CloudBurst Appliance

IBM WebSphere CloudBurst Appliance (WCA) (如图12.12所示)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。

WCA的组成主要分为三个基础性的核心部分：器件、云和虚拟系统。器件包括硬件、管理软件和一系列的预先安装和配置好的虚拟镜像和模板；云提供服务器虚拟化能力、网络能力和存储能力给器件使用，WebSphere应用运行在云中；虚拟系统将虚拟镜像和模板进行可定制化转换，以便将WebSphere应用服务器虚拟系统部署到云中。

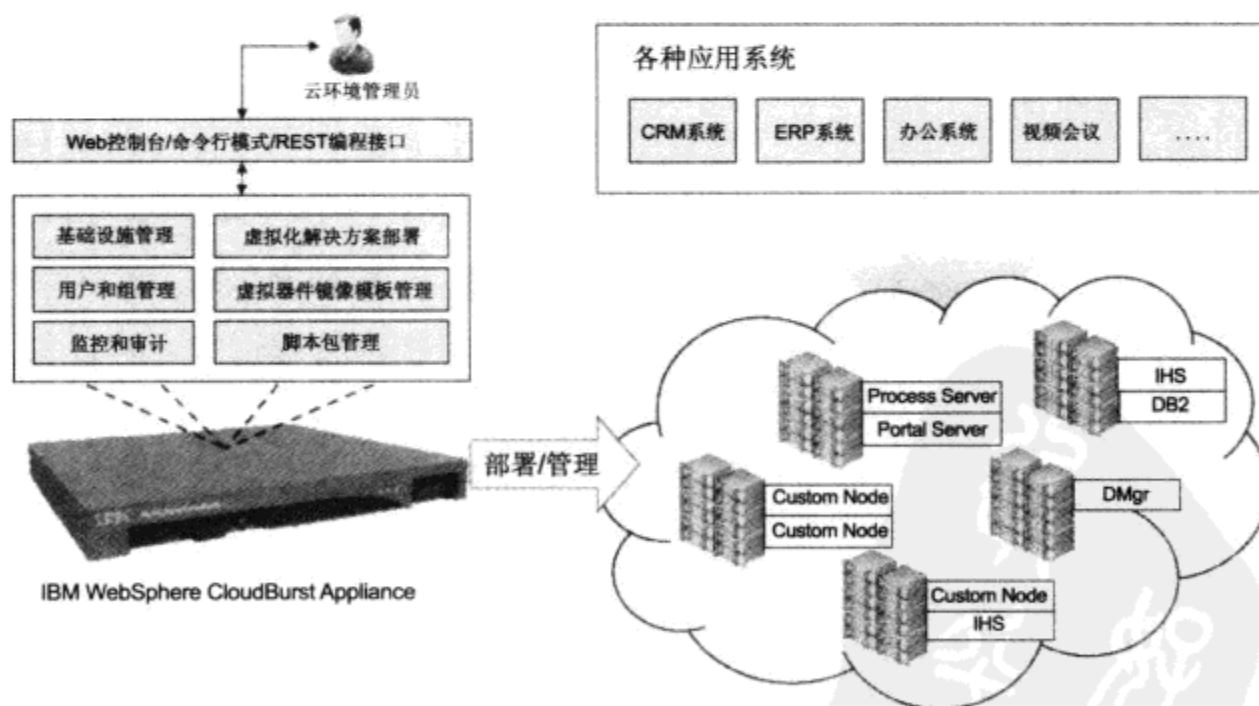


图12.12 IBM WCA

从图12.12可知，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），其中包含了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、监控与计费及脚本包管理等功能。

- ▶ **基础设施管理：**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。
- ▶ **解决方案部署：**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。这个过程有点类似于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。
- ▶ **用户群组管理：**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。
- ▶ **镜像模板管理：**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，目的是为了方使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。

- ▶ **监控与计费：**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。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。
- ▶ **脚本包管理：**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。

除了上述六大功能，WCA还支持许多其他基础性的功能。WCA支持用户通过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；WCA还支持用户日志的记录功能，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；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。

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。

WCA的使用方式很简单，即插即用的硬件加上灵活的多样性的软件控制。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。

用户可以通过三种软件控制的方式访问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风格的控制台界面，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。

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（Open Virtualization Format，开放虚拟机格式）格式打包，保证了兼容性和标准化。

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，减少服务器和中间件管理的成本，显著节约企业用户的投资和运营成本，帮助用户实现更高的投资回报率。

12.1.11 IBM System Director

现代企业对IT系统容量和性能的要求越来越高，IT基础设施日益复杂，IT管理难度和管理成本随之提高。如何有效保持关键业务系统的高可用性，减低故障率，节约运营成本，达到高效管理，是企业面临的最大的挑战之一。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，将不同硬件设备（包括服务器和存储等），不同的操作系统和不同虚拟环境进行集中的管理，才有可能简化IT管理，降低IT管理的复杂性和管理难度，通过提高IT管理效率和提高系统使用率来降低成本。由此，IBM推出了IBM System Director。

IBM System Director是一套平台管理软件。它能够帮助企业轻松有效地管理物理设备和虚拟IT环境，为客户提供了简单易用的方式来对不同硬件设备，不同的操作系统和不同虚

拟环境进行集中化的管理，帮助客户缩短故障排除时间，提高IT管理效率和系统使用率来降低管理成本。

如图12.13所示，IBM System Director由三个主要的组件组成：一个单独的基于Web的控制台，一个中央管理服务器，以及安装在被管理端的代理程序。基于Web的控制台提供了多任务的统一界面，简化了管理，同时提供了可扩展的高级特性。

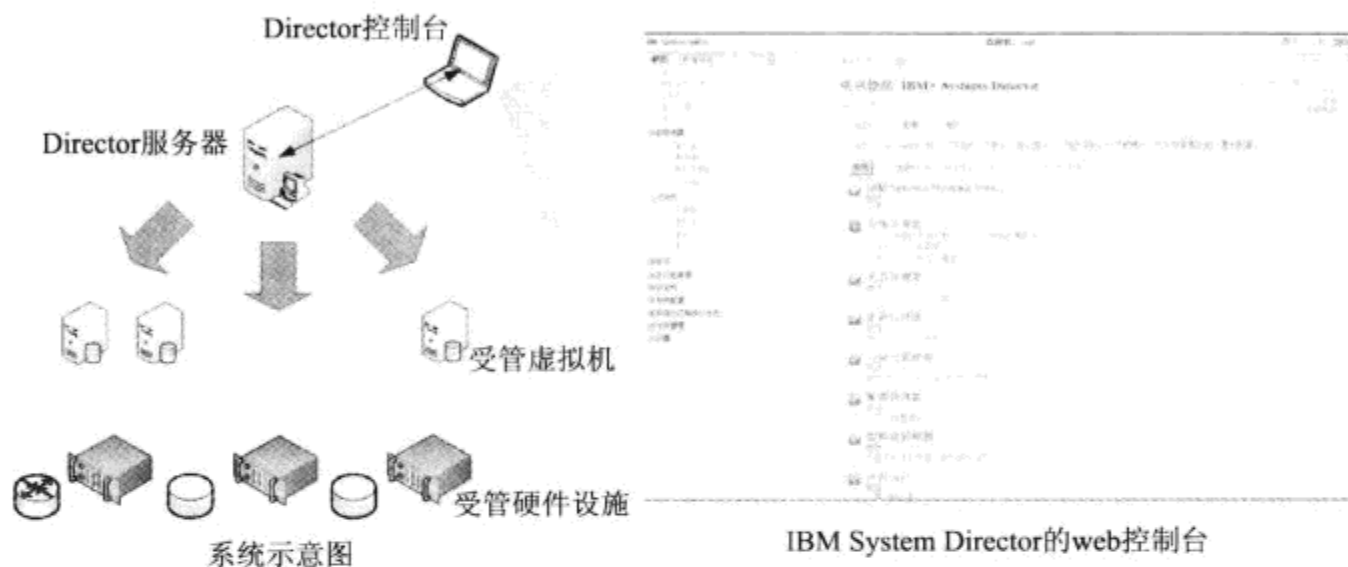


图12.13 IBM System Director

1. 基于Web的控制台

IBM System Director控制台是基于统一Web界面的，因此管理员可以从任何支持的浏览器来访问，非常简单。这种灵活而直观的基于Web的界面可以帮助客户完成各种各样的日常任务，包括安装和部署、搜索任务和资源、动态跟踪系统状态、可视化对应资源之间的关系图、查看被管理系统的运行状况等。

2. 中央管理服务器

IBM System Director中央管理服务器提供了诸多的核心服务，主要包括一系列管理器：发现管理器、状态管理器、自动化管理器、配置管理器、虚拟化管理器、更新管理器、扩展插件以及平台管理器。

(1) 发现管理器

集合了系统发现和库存收集的一系列功能，管理员可以使用交互式界面进行基本的系统发现，也可以通过易用的向导功能进行高级系统发现。可发现的内容包括：在复杂环境里发现各种物理和虚拟服务器、存储设备及网络设备；通过收集库存来查看系统数据，比

如物理、逻辑或虚拟硬件，软件应用程序，操作系统，中间件，固件与BIOS，诊断以及网络信息；通过管理概要文件来设置发现库存的标准。

（2）状态管理器

提供被管理系统的监控状况（包括硬件、操作系统、应用程序和安全），管理员可以很容易地使用状态管理器作为一种诊断手段，通过自定义的层次查看系统的健康情况。状态管理器支持诸多状态管理的功能，主要包括：查看被发现系统的重要状态，并帮助用户管理系统状况；总结运行状况，用记分板和仪表板来表示环境中被管理资源的运行状况和性能；支持查看问题和事件日志，以识别问题并确定故障根源；支持定义监视器和阈值来监控被管理系统的动态属性。

（3）自动化管理器

提供迅速准确的应急处理机制，实时地监控系统并且在系统发生硬件或者资源异常时（比如风扇损坏停转，CPU温度过高，内存使用率过高等）自动地启动预先定制的相应操作，比如发送SMTP邮件，启动程序，重启系统等。自动化管理器提供的主要功能有：提供二十种不同的操作类型支持用户创建事件操作；预定义了几十种用户环境中常见的硬件和软件事件支持用户创建时间过滤器；提供自动化计划向导以使用户能够容易地创建自动化计划；提供调度任务的功能，可单次执行或循环执行计划任务。

（4）配置管理器

支持用将新的硬件设备集成到管理环境中，进行安装后的系统配置，或者为特殊的要求做一次配置。管理器提供精心设计的适用于各种系统的一系列模板，提供了统一的用户体验，来配置服务器、存储、网络资源。配置管理器提供的主要功能有：集中浏览系统配置状态和查看配置任务概要；为一个或多个系统（各种硬件或者操作系统）的部署、分配和使用，执行初始化配置；使用配置计划的自动部署功能，自动配置新发现的系统；重新配置系统，以备重新部署、分配或者重新指配；实时修改配置，并保存配置信息至配置模板中；管理配置模板和配置计划。

（5）虚拟化管理器

虚拟化技术已经开始在企业领域广泛应用，基于虚拟化技术的解决方案可以提高CPU、内存、存储设备等硬件资源的利用率，使您可以实时、动态地规划这些资源。IBM

System Director正是IBM基于下一代虚拟化企业信息化结构在系统管理方面提供的强有力工具。虚拟化管理器是IBM System Director的虚拟化管理组件，它实现了对虚拟化资源的创建、编辑、重定位和删除功能。虚拟化管理器不仅支持包括硬件管理控制台（HMC）、集成虚拟管理器（IVM）、z/VM（部分支持）在内的IBM的虚拟化环境，并且广泛支持当今流行的虚拟化环境，比如微软虚拟服务器（Microsoft Virtual Server）、VMware和Xen。用户可以通过虚拟化管理器的单点接口，完成企业硬件和虚拟化资源的统一管理。

虚拟化管理器的主要功能包括以下几个方面：支持当今流行的虚拟化环境；一目了然的拓扑结构图，可以清晰、便捷地查看物理资源与虚拟资源之间的对应关系；实时追踪虚拟资源的状态和各种告警信息，可以快速定位问题；灵活创建可计划、可自动执行的任务来配置资源，比如基于某一种硬件资源的预警事件来重定位虚拟服务器；完整的虚拟化任务支持，比如在某个物理服务器上创建、编辑、重定位虚拟服务器。

（6）更新管理器

旨在确保受管系统所需的软件和固件始终具有最新的更新状态。通过定制系统一致性策略，可以实时监控受管系统更新状态和自动接收更新提醒。同时，更新管理器还提供了一站式更新安装操作，下载安装一次完成，无须手动干预；支持多种更新方式，既可将某个特定更新一次部署到多个系统上，也可以同时将多个不同更新安装到多个系统上，满足用户不同更新需求。

（7）扩展插件

除了最基本的功能以外，有些企业的业务有一些比较特殊的需求，比如服务和支持、能源管理、虚拟映像管理等。针对这些需求，用户可以在IBM System Director的基本平台上安装一种或多种所需要的扩展插件，比如服务和支持管理器和能源管理器。

（8）平台管理器

对IT架构中的各种系统的整个服务生命周期提供支持。通过平台管理器收集的信息，能够被IBM System Director的其他管理器获取、显示和使用，以帮助系统管理员及时了解系统的变化，从而确定下一步的操作。作为平台管理工具，IBM System Director对于各个平台，都提供了具体的平台管理器，并且都提供了如下的功能：浏览被管理系统的状态和性能；系统发现、系统库存查看、配置和更新；创建组，并针对一组平台运行任务。IBM System Director提供的具体平台管理有：IBM System x及刀片服务器的平台管理；Power系统

管理；存储管理等。

3. 管理端的代理程序

IBM System Director可以管理以下几种类型的代理：安装了公共代理程序的系统，安装了平台代理程序的系统，以及未安装任何代理程序的系统。管理员只需要在网页界面上简单地运行一个任务，就可以把代理程序或者子代理程序安装到想要安装的系统上。IBM System Director对安装了公共代理程序的系统提供以下功能：系统的发现和完整库存信息的收集、系统状态的监测和报警、远程部署和安装、远程控制，包括文件传输、远程电源的管理、资源和进程监测，设置报警阈值、驱动及代理程序的更新；使用代理管理器来统一管理所有代理的安全认证；使用了一个永不中断的进程来监视代理的状态，使其保持一直工作的状态。IBM System Director对安装了平台代理程序的系统提供的功能是公共代理程序的一个子集，包括系统的发现和部分库存信息的收集、系统状态的监测和报警，以及远程部署和安装。

总之，IBM System Director为客户提供了简单易用的方式来对不同硬件设备（包括服务器和存储等），不同的操作系统和不同虚拟环境进行集中的管理。通过简单统一的Web界面，客户可以管理底层硬件、操作系统和平台管理器。由于IBM System Director是基于行业标准开发的，它既能管理IBM的平台，又能管理非IBM的平台。

12.2 Amazon

Amazon公司成立于1995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同时增加公司的收入，Amazon公司推出了一系列云计算Web服务。本节将先对这些服务进行概述，然后介绍其中最主要的几个服务。

12.2.1 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，也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目前提供的产品如表12.1所示。

表12.1 Amazon AWS产品分类列表

产品分类	产品名称
计算	Amazon Elastic Compute Cloud (EC2)
	Amazon Elastic MapReduce
	Auto Scaling
内容交付	Amazon CloudFront
数据库	Amazon SimpleDB
	Amazon Relational Database Service (RDS)
电子商务	Amazon Fulfillment Web Service (FWS)
消息通信	Amazon Simple Queue Service (SQS)
	Amazon Simple Notification Service (SNS)
监控	Amazon CloudWatch
网络通信	Amazon Virtual Private Cloud (VPC)
	Elastic Load Balancing
支付	Amazon Flexible Payments Service (FPS)
	Amazon DevPay
存储	Amazon Simple Storage Service (S3)
	Amazon Elastic Block Storage (EBS)
	Amazon Import/Export
支持	AWS Premium Support
Web 流量	Alexa Web Information Service
	Alexa Top Sites
人力服务	Amazon Mechanical Turk

AWS基础设施层服务包括表12.1中的计算服务、消息通信服务、网络通信服务和存储服务。一个应用的生命周期典型地涵盖了创建、部署、运行、监控和卸载等几个阶段。图12.14显示了在一个应用中经常使用的各个AWS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。

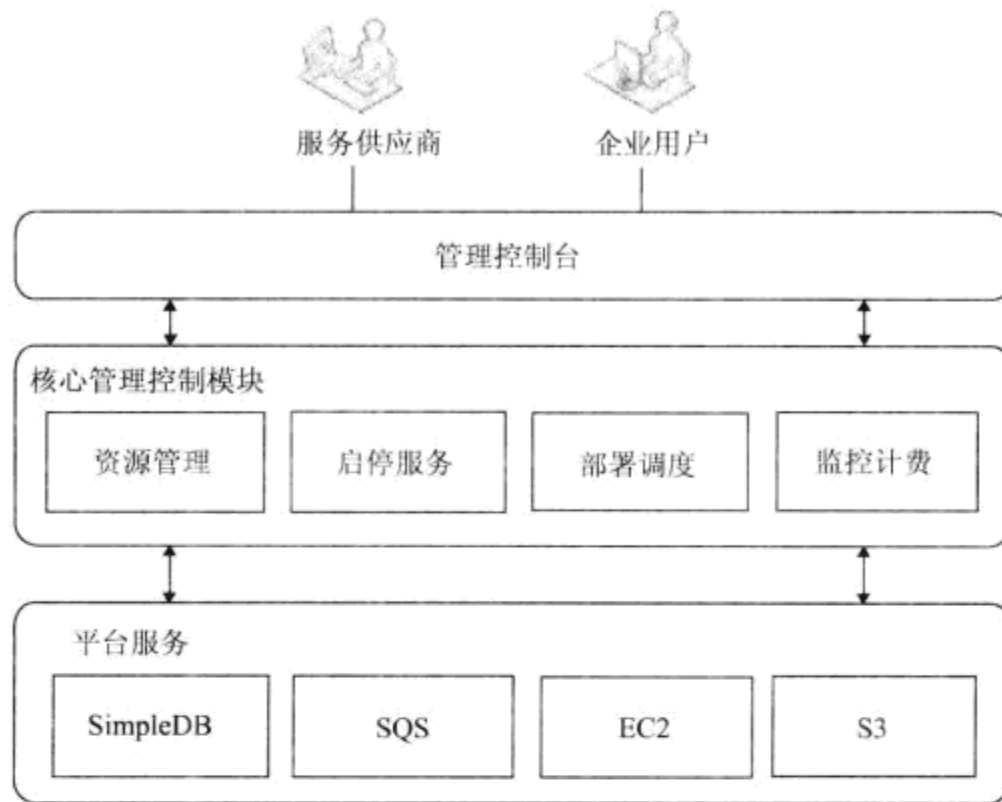


图12.14 应用所用到的AWS服务举例

AWS平台层服务不仅能够满足很多方面的IT资源需求，还提供了很多上层业务服务，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 and 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。本章下面几节将分别介绍S3，SimpleDB，RDS，SQS和EC2等几个底层关键产品。

12.2.2 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。

12.2.3 Amazon SimpleDB

Amazon SimpleDB是一种高可用的、可伸缩的非关系型数据存储服务。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，因而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。

12.2.4 Amazon RDS

尽管SimpleDB提供了一种简单、高效的数据存储服务，但是当前很多已有的应用多数还是采用关系型数据库进行数据存储，这就增加将这些应用系统迁移到Amazon AWS平台的成本和技术风险。因此，Amazon又推出了Relational Database Service (RDS)来满足用户对关系型数据库服务的需求。

RDS是一个关系型数据库服务，通过RDS用户可以非常容易地建立、操作和伸缩云中的数据库。RDS为用户提供了一套完整的MySQL数据库服务，这就使得那些目前正在使用MySQL数据库的应用可以无缝地与RDS进行集成。

毫无疑问RDS是弥补了Amazon在关系型数据库服务领域的一个空白。然而，这并不意味着RDS出现之前用户就没有办法在Amazon EC2上使用关系型数据库，也并不能意味着RDS出现之后就能满足所有应用对关系型数据库的需求。

首先，在RDS出现之前，用户可以选择将数据库产品打包在AMI镜像中并部署在EC2上运行，然后在应用中直接去对数据库进行访问。

另外，根据最近5年的Gartner的数据统计，IBM、Oracle和Microsoft的数据库产品几乎占了市场占有率的80%以上，而RDS目前只能提供对MySQL的完整支持。因此如果RDS要获得巨大成功，在未来的版本中不得不考虑如何为更多的客户提供对主流数据库产品的完整支持，比如IBM DB2。

12.2.5 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，从而创建可靠的大规模的分布式系统。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，先送入队列的消息也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，获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12.15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。

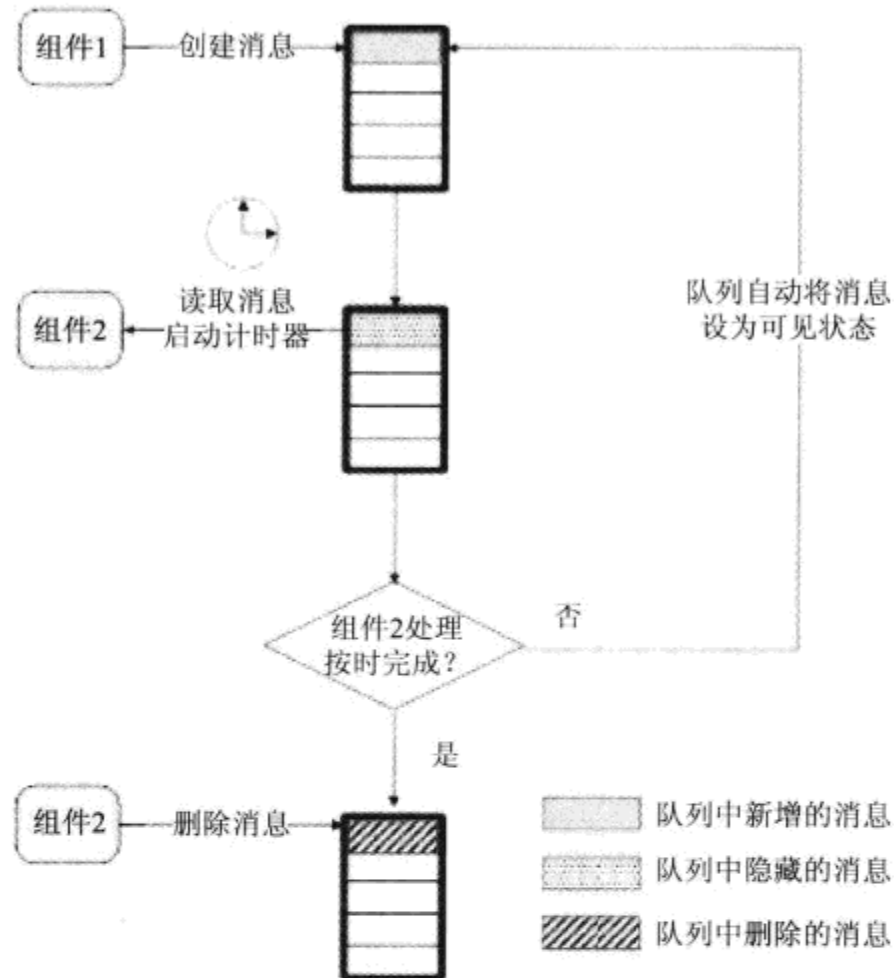


图12.15 Amazon SQS服务消息管理示例

12.2.6 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、可伸缩的计算运行环境。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灵活地申请或者终止资源使用，且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，用户可以灵活地开启、停止、增加、减少虚拟机实例，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。

12.3 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。

12.3.1 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12.16所示。

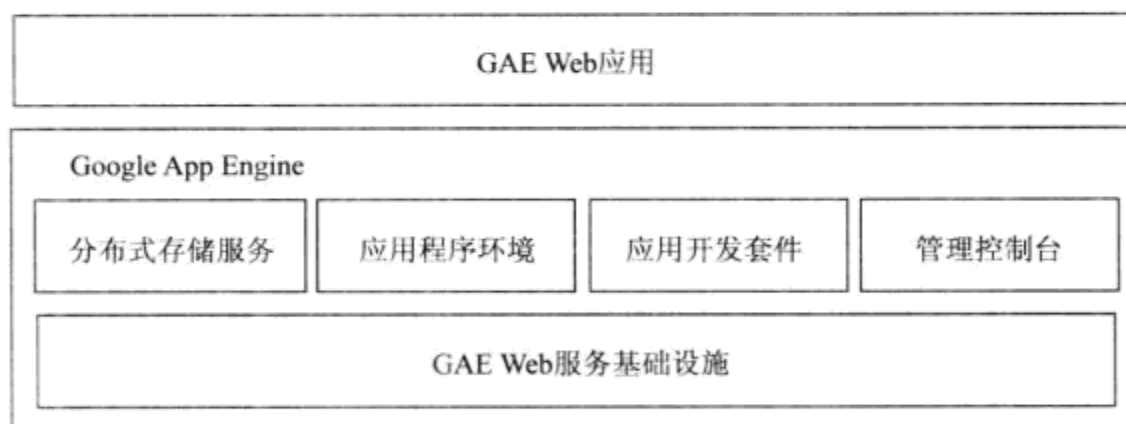


图12.16 Google App Engine系统结构

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。

12.3.2 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值实体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；另一种是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。

12.3.3 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Java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Java6，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。

12.3.4 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上传应用程序，其包含的Eclipse GAE插件能够极大地简化在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。

12.3.5 云应用

Google公司的云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。本节着重介绍Google Docs和Gmail这两个云应用。

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。

12.4 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。

12.4.1 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主要向企业用户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软件供应商在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。

12.4.2 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10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。

12.4.3 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添加、删除、查询、修改。下面将介绍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到相应的列上。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添加、删除、查询、修改和其他管理操作，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添加、删除、查询、修改权限，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。

12.4.4 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。

12.4.5 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，因此非受控包适合于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的都可以创建。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。

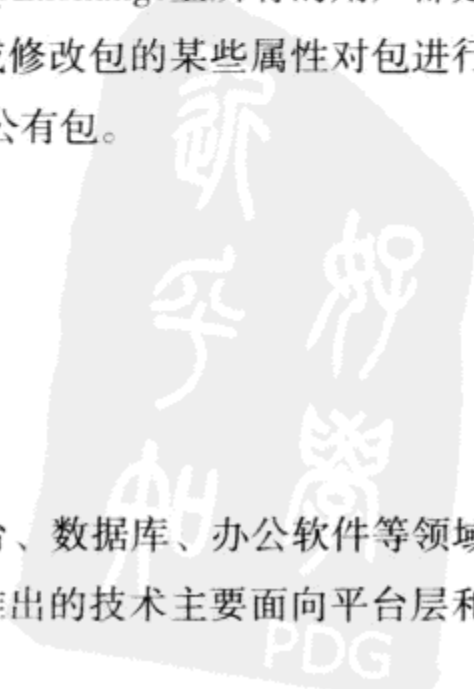
对于受控包的创建，Force.com提出了严格的要求：第一，开发者必须具有Developer Edition的身份；第二，为了防止和其他受控包冲突，开发者必须在Force.com注册命名空间前缀（Namespace Prefix）。在给受控包添加完内容项之后，开发者需要注册命名空间前缀，并且指定刚才创建的受控包，保存以后，Force.com会提示受控包创建成功。

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（ManagedRelease）。

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。

12.5 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，Microsoft推出的技术主要面向平台层和应



用层。在平台层，该公司于2008年10月正式推出了云计算产品Windows Azure platform。Windows Azure platform是运行在Microsoft数据中心，为互联网用户提供的一组云计算服务的集合，这些服务可以被单独使用，也可以被组合使用。在应用层，Microsoft提供Live服务，包括提供Internet应用的Windows Live和提供办公应用的Microsoft Office Live。

下面分别介绍这些产品和服务。

12.5.1 Windows Azure platform

1. 概述

Windows Azure platform运行在微软的数据中心中，它即是云计算应用的开发平台，又是部署平台，同时也是微软的云计算运营平台。Windows Azure platform由Windows Azure及一组平台层服务构成。Windows Azure platform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 platform给云应用层提供的平台层服务包括：（1）Windows Azure platform AppFabric，为基于.NET的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。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 platform平台中的核心组件。

2. 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，对构建高效、可靠、可伸缩的云计算平台起着重要的作用。Windows Azure由三部分组成：Fabric Controller、存储服务、计算服务。Fabric Controller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 and 存储等资源组成一个逻辑的资源池，统一管理云中资源。存储服务和计算服务是Windows Azure对外提供的服务，它们相互独立，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。开发人员通过使用开发环境，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中的几个关键部分：

Fabric Controller、存储服务、计算服务。

Fabric Controller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。应用被上传到Windows Azure platform后，Fabric Controller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 Instance服务器，以及需要多少虚拟机实例运行Worker Role Instance服务器等。

应用部署完成后，Fabric Controller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 Controller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 Controller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 Controller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 Controller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。

在存储服务部分，Windows Azure提供的不是一个关系数据库系统，而是提供了针对四种数据结构的存储服务以满足应用的不同需求，这四种数据结构是Blob、表（Table）、队列（Queue）和驱动器（Drive）。Blob存储服务能够支持用户存储数据量大的数据集合，例如图片文件、声音文件等。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。通过采用虚拟机进行规模伸缩，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，例如用于计算服务中Web Role Instance和Worker Role

Instance的通信。而Drive数据结构是为应用程序和用户提供一个NTFS文件卷，使得应用程序和用户可以通过NTFS API来进行文件操作，这样用户可以方便地将已有的基于文件操作的应用迁移到Windows Azure中。

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（64位版本），而虚拟机管理软件是基于Microsoft公司的Hyper-V针对云计算环境进行改造来实现的。Microsoft公司目前提供的Windows Azure版本支持三种虚拟机类型：一种是Web Role Instance，负责接收客户端的HTTP请求；一种是Worker Role Instance，负责从Web Role Instance或者存储队列中接收输入和执行计算，并将计算结果返回给Web Role Instance或者写到指定的存储位置。为了支持应用的伸缩，Web Role Instance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 Instance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 Instance而不会对应用的运行产生影响。第三种是VM Instance，类似于一个标准的虚拟机环境，从而使得用户可以直接对虚拟机进行控制和管理，提供很高的灵活性和通用性，用户也方便把现有的应用非常方便快速地迁移到Windows Azure环境中。

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 Instance和Workers Role Instance），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 Platform上，对外提供服务。

3. Microsoft Azure Platform AppFabric

Microsoft Azure Platform AppFabric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.NET 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。Microsoft Azure Platform AppFabric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），这两个模块一个保证了应用之间连接时的安全，一个保证了应用之间的信息的共享和传递。AppFabric还有更多的模块即将推出。

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。

4. 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高可用的、可伸缩的、多租户的、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 Azure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, 无须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服务的应用能获得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。

12.5.2 Live服务

1. 概述

Live服务在Windows Azure平台产生之前就已经存在了。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化应用，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。在Microsoft对于Windows Live的2011年规划中，Live Mesh与另外一个服务Live Sync进行了合并，合并后的服务将被称为Live Sync 2011。

2. Windows Live

使用Live服务的最著名应用是Windows Live，其中集成了Live Messenger、Live Hotmail、Live SkyDrive、Live Alerts、Live Calendar、Live Contacts、Live Groups、Live Photos、Live Spaces等常用的基于Web的应用。下面介绍几个常用的Windows Live服务。

Live Messenger是Windows Live中最著名和使用率最高的应用，它为用户提供消息即

时通信服务，用户使用Live Messenger有多种方式，可以通过在本地操作系统上安装Live Messenger客户端，也可以通过Web浏览器访问Live Messenger的Web版本。Live Messenger会保存用户的联系人列表，并支持对联系人进行分组。通过Live Messenger客户端还可以进行文件传输、语音聊天、视频会议等操作。

Live Hotmail是Windows Live中另一个最为知名的应用，它是一款在线的电子邮件应用，自1996年以来持续提供服务。用户通过Live Hotmail，可以不需安装邮件客户端，而在网页上收发邮件。Live Hotmail提供了丰富的文本编辑功能，使得用户在网页上也可以编写出形式多样的邮件。Live Hotmail通过访问在线的垃圾邮件数据库，可以智能地为用户过滤垃圾邮件，从而避免了垃圾邮件带来的安全隐患。

Live SkyDrive是Windows Live中的在线存储服务，即通俗所说的“网络硬盘”。用户可以通过浏览器将本地文件上传到SkyDrive中，或将SkyDrive中的文件下载到本地。SkyDrive还能够很方便地使文件在多个用户之间进行共享。SkyDrive还有一个非常贴心的功能，当用户想下载整个文件夹的全部文件时，SkyDrive可以将这些文件打包成一个ZIP包，使用户只需下载一次即可。

Live Spaces是Windows Live中的社会网络服务，用户可以在Spaces中发布自己的博客、照片、音乐，访问好友的Spaces，相互进行评论。Spaces中的博客编写同样可以支持所见即所得的方式，从而为用户提供了很大的便利。

Windows Live中的这些服务并不是相互孤立的，而是通过Windows Live ID作为识别来进行协作。每一个Windows Live的用户都具有一个唯一的Windows Live ID，作为使用Windows Live服务的识别码。例如，Live Messenger、Live Hotmail、Live Spaces这几个应用可以共享Live用户的个人资料和联系人信息。

广义上讲，Microsoft提供的搜索服务Bing也是Windows Live的一部分，它取代了原有的Windows Live服务Windows Live Search。通过Bing，用户可以搜索Internet上的网页、图片、视频等。Bing还提供了类似Google Map的地图服务，以及电子词典服务。

3. Microsoft Office Live

Microsoft Office Live是Microsoft Office的在线版本，其目标是让用户在Internet上像在自己的机器上一样使用Office的软件，编辑Office文件。Microsoft Office Live目前还在开发阶段，目前可以使用的服务是Microsoft Office Live Workspace（Beta版本）。

Microsoft Office Live Workspace为用户提供了一个保存自己的Office文件的网络空间，用

户可以从任意能接入Internet的机器通过自己的Windows Live ID登入Office Live Workspace, 然后在其中上传、浏览、管理、共享、下载自己的Office文件, 当本机安装了Microsoft Office软件时, 用户只需在Workspace中选择新建或者编辑文件, 就可以自动打开本地的Office软件进行编辑, 用户选择储存文件时, 文件会自动在Workspace中储存。用户还可以为本地的Office软件安装Office Live Update 1.5插件, 使得用户可以从Word、Excel、PowerPoint的菜单中直接连接Office Live Workspace。

Office Live最新的进展是Office Web Apps, 目前作为实验技术出现在Office的最新版本Office 2010中。Office Web Apps的目标是让用户在只需具备Internet连接和浏览器的设备上(台式机、笔记本电脑、智能手机等)对Office文档进行浏览和基础的编辑, 而不需在本机上安装Office软件。对于PowerPoint幻灯片文件, 还可以在浏览器中进行播放。也就是说Office Web Apps提供的是类似Google Spreadsheet的使用体验。同时, 当用户需要使用网页上没有提供的高级功能时, Office Web Apps也可以象Office Live Workspace一样, 将网络上存储的Office文件通过本机的Office软件进行处理。Office Web Apps为用户带来了明显的好处, 用户的Office文件将不再与自己安装Office的版本和操作系统平台有非常强的绑定, 这样便于多人的协作, 为此, Office Web Apps还在Excel Web App和OneNote Web App中提供了协同文档编辑功能。

当Office Web Apps测试足够成熟后, 会开放给整个Live社群, 任何拥有Windows Live ID的用户将都可以使用。同时, Office Web Apps还可以使用在企业内部环境中, 通过在SharePoint 2010服务器上安装Office Web Apps服务, 即使可以使企业网内的用户使用Office Web Apps。

12.6 开源云计算

开源软件一直都作为软件领域不容忽视的一股力量推动着整个软件业的前进和发展, 其影响范围涵盖从操作系统、中间件到应用软件等的方方面面。在云计算领域, 开源的力量更是不容忽视, 著名的产品有基础设施云产品Nimbus, Eucalyptus及平台即服务云产品AppScale等。下面将以Eucalyptus为例进行详细介绍, 而对Nimbus和AppScale进行简单的介绍。

12.6.1 Eucalyptus

Eucalyptus作为一个开放的基础设施云系统, 为IT管理员和学术研究提供了一个开放的平

台。通过Eucalyptus用户能够快速利用已有的集群资源或者工作站搭建一个弹性的云计算系统，管理虚拟机、存储及网络基础设施资源。这个项目起源于加州大学圣塔芭芭拉分校（UCSB）计算机学院的一个研究项目，最初的目的是为了加深用户对云计算的理解和认识，为云技术爱好者提供一个同公有云兼容的实验系统，同时能为开源社区提供一个基础软件开发系统，凝聚开源社区的力量不断完善。经过不断地发展，现在Eucalyptus已经尝试商业化，由Eucalyptus System公司管理。Eucalyptus System公司在开源版本的基础上提供增强的功能和服务，致力于提供企业级的云计算方案。与此同时，在开源社区，Eucalyptus仍按照开源软件开发和维护。

Eucalyptus采用了模块化、可移植性的设计准则，不仅能和已有的基础设施资源兼容，而且简单易用，使用户可以快速安装和配置。模块化的设计使用户可以根据需要对系统的功能进行定制、更改或者替换。可移植性保证了可以运行在当前主流的Linux系统上，摆脱了对系统的限制，使用户可以使用已有的资源，而不必重新采购硬件，降低了用户迈入云计算的门槛。Eucalyptus的另外一个亮点是兼容Amazon EC2以及S3的接口，所以基于Eucalyptus系统的虚拟机镜像及软件能快速迁移到Amazon系统上运行，熟悉Amazon系统的用户也可以利用Amazon提供的工具同Eucalyptus系统进行交互。Eucalyptus架构如图12.17所示。

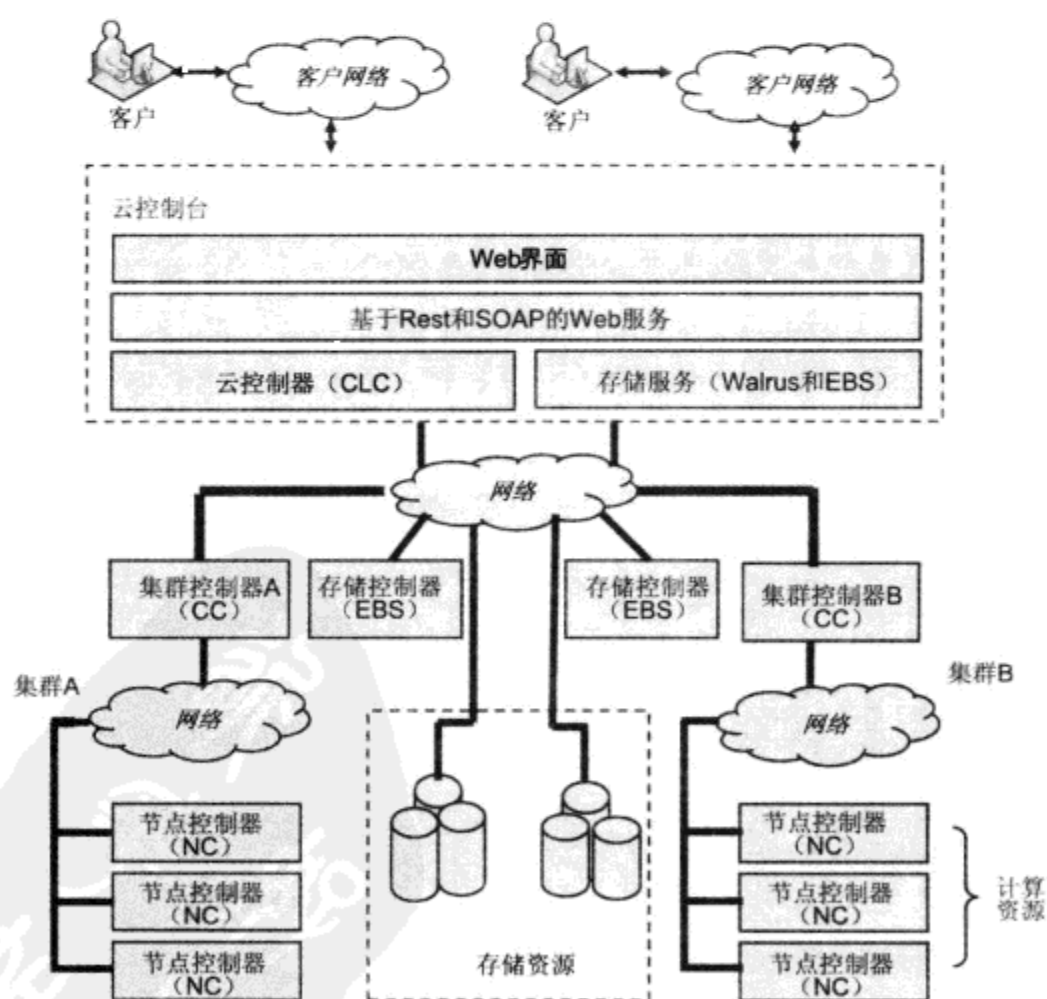


图12.17 Eucalyptus架构

在图12.17中，Eucalyptus主要由以下四个核心组件构成：节点控制器（NC）、集群控制器（CC）、云控制器（CLC）及存储服务（EBS和Walrus）。节点控制器安装在每个被管物理节点上，负责监控节点的资源使用情况和管理虚拟机的生命周期，如虚拟机实例的创建、监控、配置及删除。一组节点组成一个集群，每个集群拥有一个集群控制器和一个存储控制器；集群控制器负责采集集群中节点的信息，在集群范围调度虚拟机的执行及进行网络管理；而存储控制器负责管理实例的EBS（弹性块存储，概念来源于Amazon）类型存储，为虚拟机实例提供持久化的块数据存储服务。在集群控制器之上便是云控制器，云控制器是系统管理员和终端用户提供的访问系统的入口，接收用户的资源请求，然后通过集群控制器搜集节点的资源信息，进而在集群之间进行调度。存储服务包括Walrus和弹性块存储服务，其中Walrus实现了Amazon S3的服务接口，支持用户数据和虚拟机镜像的上传、下载和查询；弹性块存储服务为虚拟机实例提供持久化的块数据存储。存储控制器可以同集群控制器一同为虚拟机实例提供弹性块存储服务。

节点控制器部署在每个物理节点上，通过和操作系统或者Hypervisor协作来完成资源信息的采集和对虚拟机的控制。目前支持Xen及KVM的Hypervisor，在将来会支持更多的Hypervisor。节点控制器响应来自集群控制器的查询和控制请求，目前主要包括两种类型请求：资源查询请求和虚拟机实例查询控制请求。节点控制器通过基于脚本的发现机制发现Hypervisor的CPU、内存及磁盘资源，然后返回给集群控制器。节点控制器通过调用Hypervisor的接口或者执行命令获得节点的所有虚拟机实例的状态及资源使用情况，控制虚拟机的启动和终止，完成集群控制器的控制请求。

除此之外，节点控制器还负责对用户进行权限认证，只有虚拟机实例的拥有者才能够实现对其进行访问和控制。启动新虚拟机实例包括如下几个步骤：首先从本地或者虚拟机镜像库复制虚拟机镜像到目标节点；然后是配置虚拟机网络，包括对虚拟机进行分组以及防火墙设置等；最后是调用Hypervisor接口或命令启动虚拟机。停止虚拟机实例则要首先调用Hypervisor接口停止虚拟机；然后是删除虚拟机实例网络及本地所有相关文件。

集群控制器部署在集群的某一个节点上，不仅负责集群范围内资源和虚拟机管理，而且负责集群内部网络的管理。在资源和虚拟机管理功能上同节点控制器类似，差别是在更高的层次，主要包括：负责集群内所有节点的资源信息采集和汇总，控制集群内的虚拟网络，以及在节点控制器之间调度虚拟机实例创建请求。通过和集群内的每个节点控制器进行通信从而获取集群内部节点的详细信息，包括每个节点的可用资源，虚拟机实例的状态

等。当接收到虚拟机实例创建请求时，集群控制器根据每个节点的可用资源情况在节点间进行调度；当接收到云控制器的资源描述请求时，集群控制器会根据请求中的资源要求和集群内可用资源的情况计算出集群可支撑的虚拟机实例数量，返回给云控制器。网络控制是集群控制器的另外一个重要功能。

网络设计无疑是基础设施云中最有挑战性的工作之一，面对大规模的虚拟机实例，怎样保证虚拟机网络的性能，同时实现同一组虚拟机之间的网络连通和不同组之间虚拟机的网络隔离显得尤为重要。Eucalyptus当前提供四种网络模式：系统模式、动态模式、被管模式和不支持VLAN的被管模式；系统模式是在虚拟机实例启动前系统为虚拟机分配一个随机的MAC地址然后利用以太网网桥配置虚拟机和物理网络的连接，实例启动后DHCP服务器为虚拟机实例分配IP地址；静态模式需要管理员预先配置MAC和IP地址对的网络资源池，当虚拟机启动后便通过由Eucalyptus控制的DHCP服务器为虚拟机获取MAC及IP地址；上面两种模式有以下不足：无法对虚拟机之间的访问规则进行控制，不能够对虚拟机之间的通信进行隔离及实现虚拟机实例IP地址的动态绑定。针对这些问题，Eucalyptus提供了两种更高级的网络配置模式。被管模式是通过防火墙配置对不同安全组（用户可以通过配置将虚拟机实例分组）之间的虚拟机网络流量进行隔离；采用动态绑定技术为虚拟机动态的分配公网IP地址，实现IP地址在同一用户的所有资源之间进行共享。

被管模式和前两种模式相比优势在于更安全，隔离性好及配置灵活，而由此带来的网络性能的损耗是这种模式的美中不足。被管模式通过VLAN技术实现了不同虚拟机安全组之间网络的隔离，但因为所有的网络请求都会经过集群控制器，所以性能会有所降低。对于不需要网络隔离的用户来说，第四种模式即不支持VLAN的被管模式无疑是最佳的选择，它同样支持IP地址动态绑定以及安全组策略。

上面介绍了Eucalyptus的计算资源和网络资源管理，下面将介绍基础设施云中最后一个关键的模块，即存储的管理。存储管理模块负责云中所有的存储资源的管理，包括弹性块存储（EBS）服务和兼容Amazon S3服务接口的存储服务（Walrus）。EBS为虚拟机实例提供了持久的块存储设备，可以将实例的数据和状态进行持久保存。目前EBS提供对存储卷的创建、快照、删除，以及和虚拟机实例绑定等功能。

Walrus是兼容Amazon S3的REST或者SOAP访问接口的数据存储服务，用户不仅可以从互联网或者通过虚拟机实例上传、下载用户数据，还可以保存虚拟机的镜像或者快照。为了支持扩展，Walrus不提供对数据写操作加锁服务，从而当发生写冲突的时候，前次写操

作会自动失效。与虚拟机控制一样，在用户访问存储服务时系统会进行权限的验证，只有认证成功才能访问资源。

利用Walrus存储虚拟机镜像要经历下面几个步骤：首先要利用工具和用户证书进行压缩和加密；然后是切分成多个块上传到服务器；最后服务器对接收到的数据进行校验和存储。当节点请求镜像的时候，Walrus会首先进行验证，然后是解密和解压，进而进行传输，因为虚拟机镜像的大小往往是几个GB甚至几十个GB，所以这个过程将比较耗时。为了提高访问性能，Walrus维护了一个缓存，保存经常要访问的镜像。当镜像被更新时或缓存空间不足时缓存中的镜像将被替换。

云控制器包括资源服务、数据服务和接口服务三类服务。资源服务维护整个系统层次的资源状态（SRS），利用系统的集群控制器响应用户的虚拟机实例控制请求。无论是资源请求还是为虚拟机实例预约资源，系统资源状态都是必要参考信息，而且为运行的资源预约变更和实现基于SLA机制的服务提供权威的数据。数据服务用来处理系统状态和用户数据的创建、更改、查询和存储，包括安全组、密钥、网络 and 镜像等。通过数据服务，用户可以获取系统的可用镜像，资源等信息，并且能对虚拟机实例的密钥、安全组和网络进行定制。如用户可以更改虚拟机的防火墙策略，在Web服务器上启用80端口接收互联网的Web请求。除了提供编程接口，还为用户提供了云控制台，可以注册用户、下载密钥等。管理员通过控制台可以管理系统的账户，监控系统组件的状态。

除了上面介绍的功能，Eucalyptus还实现了用户管理、虚拟机镜像管理等功能，从而构成一个完整的系统。Eucalyptus的设计使得它能够运行在研究机构或者IT部门已有的环境中，帮助研究人员和管理员管理服务器、存储及网络资源，进行技术的研究和创新。Eucalyptus可以运行在Linux集群、工作站池以及服务器群等环境中。其模块设计原则为用户定制或者替换模块提供了便捷。最后，Eucalyptus作为一款基础设施云（IaaS）系统为开源社区提供了一个良好的开发系统，它不断吸引着优秀的开发人才的加入，对系统进行持续的改进和完善。

12.6.2 NoSQL

伴随着Internet发展的就是数据的爆炸式增长。现在很多互联网公司面临着一些共同的数据问题，比如：1）大规模数据：数据的大小以TB甚至PB为单位；2）并行访问程度

非常高：每天网站有着几十万甚至几百万的页面访问量；3）数据之间关系复杂：数据和数据之间有着错综复杂的关系，所有数据已经不是以前关系型数据在大多数情况下所展示的层级式关系，而是更为复杂的网状结构；4）多样性：很多互联网公司的数据来源非常多，这些数据的结构很不一样，有结构化数据、半结构化数据和非结构化数据，而且它们的用途也不同，比如用于气象预测、专利统计、金融分析和商标信息等；5）伸缩性：当数据快速增长的时候，如何非常容易地扩容并且对数据进行划分。传统的关系型数据库要满足数据爆炸式增长所带来的挑战并不是很容易。

NoSQL数据库正式在这样的背景下产生的。NoSQL的概念在2009年被提出来，通常被解释为非关系型数据库，但是现在社区里面越来越多地人接受Not Only SQL这种解释。目前NoSQL数据库通常可以分为以下几个类型：

键值（Key/Value）或元组（Tuple）模型的数据库。这类数据库提供以键值对或元组方式的存储和查询，其代表有Memcache、Redis、Tokyo Cabinet、Dynamo、Voldemort和国内的MemcachedDB。Memcache和Redis是一个基于内存的高性能Key/Value数据库，这是因为数据的读写操作都是在内存中实现的。Redis可以定期将数据异步地存储在持久化层中，另外，相对于同类型的其他NoSQL数据库，它支持的数据类型也比较丰富，比如字符串、哈希、列表和集合等，因此Redis对支持复杂的数据结构比较胜任，通常用来实现社交网络的关系描述和计数，比如，新浪微博的架构中就大量使用了Redis服务器，国外的还有著名的git库托管网站github。Tokyo Cabinet是由日本第一大社交网站mixi.jp开发的一款高性能持久化Key/Value数据库，它采用C实现，不过支持多种编程语言，比如Perl，Ruby和Java，它在架构上采用双主机的方式，即Active-Active方式，在两台服务器都可以进行读写，该架构在扩展性方面有所局限，为了解决可扩展性问题，在Tokyo Cabinet的基础上实现了Flare系统，被应用在日本的另一个社交网站green.jp中，它通过类似负载均衡的方式将数据的读写请求转发到后端的多个服务器上。MemcacheDB是由新浪网开源出来的一个项目，一个遵循Memcached协议的Key/Value数据库，后端采用Berkeley DB实现了持久化存储，新浪博客就是它的一个应用场景。

列族（Column Family）存储模型数据库。这种数据库的数据式按照行和列的方式存储的，每行也有一个键值，但与关系型数据库不同的是列族是按照“Column Family: Qualifier”的形式组织的，所以每个行和列的交叉位置都可以存储任意模式的数据，比如同一网页不同版本的内容存储在同一个列族中。Google BigTable，Apache HBase和Cassandra

是这种模型的数据库的典型代表。其中，BigTable算是这类数据库的鼻祖，它由Google设计和开发，被广泛应用在Google的互联网产品上，比如海量网页数据索引、Google Earth和Google Finance等，其相关论文发表在2006年的顶级会议OSDI上。Apache HBase便是基于BigTable的开源项目，正是由于它的开放性，很多大型的互联网企业都基于它来建立自己的数据仓库和上层应用，比如国外的Facebook，Yahoo和国内的百度和淘宝。Cassandra也是来自于Apache的开源项目，它的设计灵感来自于BigTable和Amazon的Dynamo，架构上完全采用P2P方式，去除了BigTable或者HBase架构中的中心节点，具有较好的可用性和分区容忍性，并且可以通过对节点的配置实现不同的数据一致性要求。

文档（Document）模型数据库。文档模型的数据库通常可以包含多个数据集合，在每个集合中数据是按照行的方式进行存储的，每一行的内容并不是按照列进行存储，而是可以按照某种数据模型存储任意复杂的数据，这里的数据模型可以由JSON或者XML来描述。这种数据库的典型代表是CouchDB、MongoDB和Riak。MongoDB是10gen提供的一款由C++实现的开源产品，它采用了易读性较好的JSON来标识文档结构，模式（Schema）非常灵活自由，相对于其他NoSQL数据库，MongoDB还提供了动态查询功能，这一点和关系型数据库有点类似，不仅如此，它还自己实现了一套分布式文件系统GridFS。目前，国内外一些互联网创业公司利用它来实现web站点数据的操作、缓存和实时计数，比如著名的基于用户地理位置信息的手机服务网站foursquare就是MongoDB的用户。

图数据库。图数据库采用计算机的图结构来描述各种复杂的关系，实现难度也最为复杂。目前比较流行的是Neo4J和GraphDB，它们都是以图结构的方式组织和操作对象数据的数据库，其中的对象数据可以分为节点的描述、边（节点与节点的关系）的描述和边属性的描述三大内容，这种数据库特别适合社交网络这样的应用。不过，从目前来看，它的成熟度和流行程度来讲，还赶不上前面几种类型的产品。

除了上述几种类型的NoSQL数据库，最近还出现了一些新兴类型，比如以db4o和Versant为代表的对象数据库，以Berkeley DB XML和BaseX为代表的XML数据库，有兴趣的读者可以去查阅相关的文献资料。

在本小节中，我们按照NoSQL的分类对目前流行的NoSQL产品进行了简要的概述。不过，NoSQL也还处于发展过程中，它并不能完全替代传统的关系型数据库，而是对关系型数据库一个很好的补充。

12.6.3 OpenStack

OpenStack是为了满足构建可伸缩及易实现的各种规模共有云以及私有云的开源云计算平台需求的基础架构云解决方案，由美国Rackspace公司和美国国家航空航天局在2010年7月份共同推出，是支撑Rackspace的云计算平台的开源版本。

Rackspace是仅次于Amazon的第二大公共云服务供应商，在全球当前有9个数据中心，一共有65000多台物理服务器，并且有长达十多年的服务器托管服务。。Rackspace在云平台的管理和运营积累了丰富的经验。由此看来，OpenStack同其他基础架构云方案如Eucalyptus相比更具优势。

首先，由于Rackspace有商业化平台的管理和运营经验，能够促使OpenStack的功能得到不断的完善，性能得到保证。这比缺乏商业化的大规模用户验证其它开源软件会得到更多的资金、技术和推广的保证。

其次，Openstack是Rackspace公司同美国国家航空航天局共同合作推出的解决方案，这对OpenStack初期推广无论是在资金还是在可靠性上都给予了保障。

第三，OpenStack是基于Apache 2.0的开源模式。众所周知，Apache 2.0允许商业软件可以任意使用其软件代码，而只需提及代码的原出处就可以了，从而商业厂商显然更愿意合作。这种策略从AMD、戴尔、Citrix和微软等公司已先后宣布支持OpenStack得到证明。多厂商的合作将有效建设OpenStack所需的生态系统，进一步完善OpenStack的各项功能和性能。

如图12.18所示，OpenStack主要包括如下三个组成部分：

计算模块：代号是Nova，起初是为NASA的计算处理服务而开发；是Openstack的控制器，实现数以百万计的VM管理功能，包括虚拟机的部署、运行管理以及回收。

对象存储模块：代号Swift，是存储服务组件，主要负责大规模数据对象管理，包括冗余存储管理、备份数据管理、数据容量预测等。

镜像管理模块：代号Glance，主要解决虚拟机需要的Image文件管理问题，包括虚拟机的发现、注册、以及删除等。有意思的是，Image Service提供三种Image管理方式。一是直接

使用Object Storage，二是使用Amazon的S3存储，三是通过Object Storage调用S3。

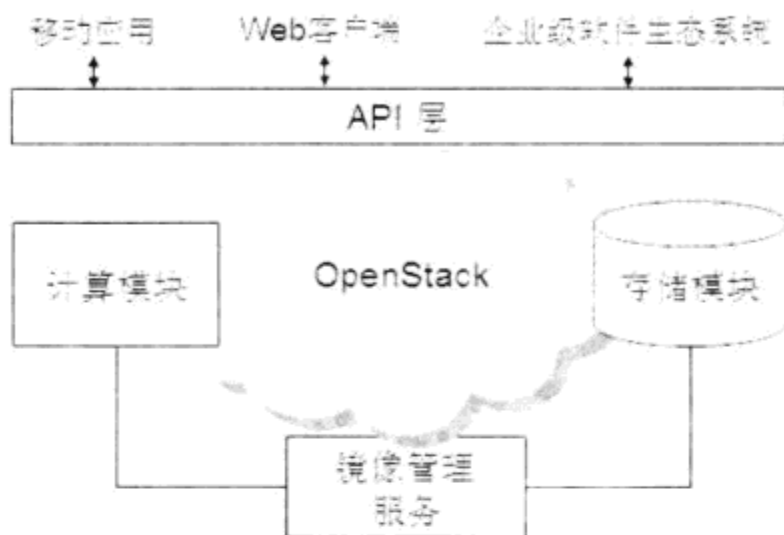


图12.18 OpenStack 功能模块图

特别值得强调的是，OpenStack支持多种X86的Hypervisor，目前，OpenStack可支持如下Hyper-V 2008、KVM、LXC、QEMU、UML、VMWare ESX/ESXi 4.1、XenServer 5.5。比起其他开源的IAAS解决方案而言，对Hyper-V的支持显然是个亮点。当然，对于构建一个完整的云计算方案，OpenStack还有一定的欠缺，比如计费，全面监控，高级的网络管理功能等还需要进一步完善或者由第三方提供。

12.6.4 Nimbus

Nimbus是网络中间件公司Globus提供的由一组开源软件组成的基础设施云解决方案，实现对虚拟机资源的管理，并以服务的方式交付给最终用户。由于Workspace Service是Nimbus的核心模块，所以最初Nimbus项目被称为Virtual Workspace Service（VWS）。Nimbus项目的主要目的是为了推动基础设施相关技术的探索以满足科学研究的需要。Nimbus架构如图12.19所示。

在图12.19中Nimbus主要由三个层次的模块组成：第一个层次由提供给客户的服务组成，包括管理虚拟机的WS服务、Web管理控制台、管理集群的Context Broker，以及提供类似S3存储服务的Cumulus。第二个层次负责对云平台中异构的资源进行管理，包括对分配给Nimbus的资源进行管理的WS资源管理模块，同其他资源管理服务或软件（如LRM）进行集成的WS Pilot模块，以及同其他云平台集成的IaaS网关模块。最下层是WS控制插件

和Context代理，其中WS控制插件被安装在每台Hypervisor上，提供对Hypervisor的操作及节点资源的管理；Context代理负责同Context Broker协作完成集群的管理。

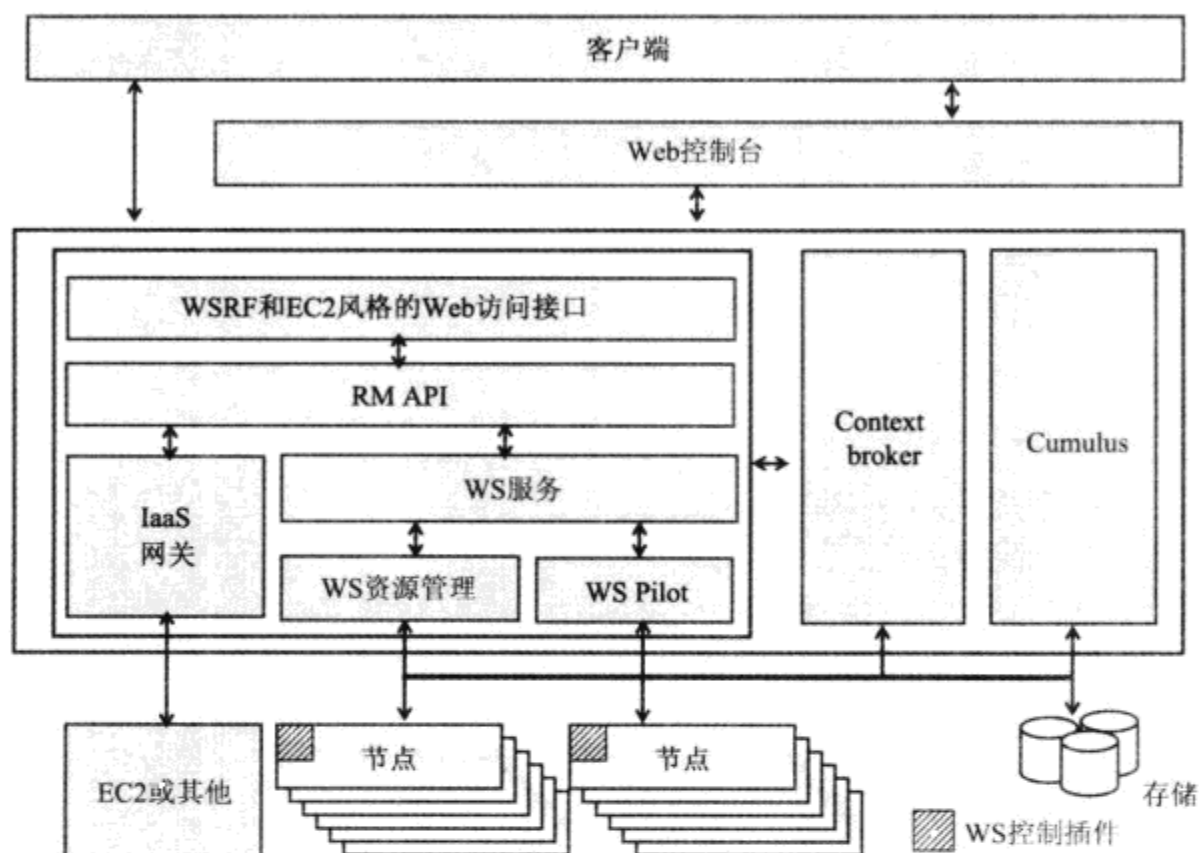


图12.19 Nimbus架构

根据设计，每个环境中会存在一个WS服务组件，支持用户通过WSRF和EC2等不同类型的访问协议远程管理虚拟机资源。WS服务的功能是通过WS资源管理以及WS Pilot模块来实现。WS资源管理和WS Pilot同安装在每个物理节点上的WS控制插件一起协作完成虚拟机生命周期管理能力，包括管理虚拟机部署请求、资源分配、运行管理及终止等。基于WS服务，Context Broker和Context代理提供集群管理的能力，如集群的创建和配置。Context代理安装在每个虚拟机中，负责接收Context Broker信息及虚拟机中执行集群的相关操作。除此之外，Nimbus还提供用户管理、用户认证和授权、虚拟机镜像管理等功能。

作为基础设施云的解决方案，同Eucalyptus类似，Nimbus提供虚拟机的管理服务和存储服务。但实现的功能又有所差异，现在版本的Nimbus（2.5版）尚不支持安全组、可用区域（Availability zones）、弹性IP管理及REST等。

12.6.5 AppScale

AppScale和Eucalyptus“师出同门”，也起源于加州大学圣塔芭芭拉分校计算机学院。

和Eucalyptus不同，AppScale专注于应用运行环境的管理，是Google App Engine（GAE）平台云的开源实现，AppScale可以运行在虚拟化基础设施云平台中，如EC2和Eucalyptus等。利用AppScale GAE的应用可以直接运行在本地的环境中，从而为研究者提供了一个研究和探索云服务、平台层和基础设施层云集成以及平台层云核心技术的基础框架。AppScale架构如图12.20所示。

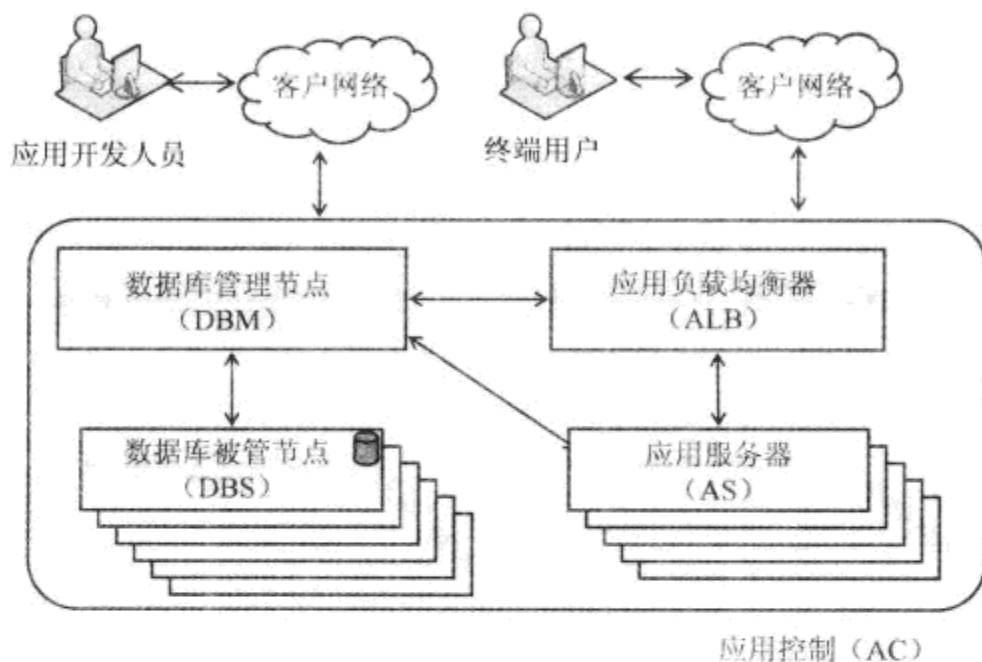


图12.20 AppScale架构

AppScale主要由六个组件组成：AppScale工具包、数据库管理节点（DBM）、被管数据库节点（DBS）、应用负载均衡器（ALB）、应用服务器（AS）和应用控制器（AC）。在AppScale环境中，DBM和ALB都只有一个实例，DBS和AS则存在至少一个实例，AC需要安装在每个组件的实例中。AC负责实例的启动、初始化、终止、管理、组件之间的通信及GAE应用的部署和认证等。AS是基于GAE SDK扩展实现的，为GAE应用提供分布式的运行环境，而DBM和DBS为应用提供可伸缩的、可靠的和分布式的存储服务，GAE应用可以通过DBM将应用的数据或状态等进行持久化。

AppScale支持两类用户角色，一类是应用的开发人员，另外一类是GAE应用的最终用户。开发人员利用AppScale Tools同平台进行交互进行AppScale环境的部署、GAE应用的提交和部署。直接同AppScale Tools进行交互的节点我们称之为AppScale的前端节点，我们通常选择ALB所在的节点为前端节点，而AppScale环境的部署和管理、资源的监控及应用的管理通常是由这个节点上的AC组件管理的。当开发人员提交一个新的应用时，AppScale首先分配一个实例并启动，而这个实例中的AC组件会随系统自动启动，然后启动节点中的

ALB, 然后初始化系统的其他组件。环境初始化后, AC便负责GAE应用的部署。在应用运行的整个生命周期, ALB所在AC节点便负责监控AppScale的组件的状态及性能, 提供一定的容错能力, 当节点失效时会启动新的节点; 当应用的负载低时自动释放利用率低的资源, 相反, 当应用的负载增加时自动增加更多的资源从而保证应用的服务质量。

作为平台层云, AppScale提供的主要功能是应用运行环境和数据存储服务。应用的运行环境由ALB和一组AS组成, ALB接收终端用户的服务请求, 进行用户认证和请求的路由, 根据路由策略将用户的请求分发到对应的应用服务器(AS), 然后用户将直接同AS进行交互直到注销用户登录。AS是对GAE SDK的扩展, 支持虚拟化分布式的环境及可伸缩、可靠、分布式的存储服务。AppScale目前为应用提供了基于HBase、HyperTable实现的存储服务接口, 其框架的灵活性保证能通过扩展来支持更多的存储机制或产品。

AppScale是一个同基础设施云整合的平台层云方案, 能够使GAE应用不加修改地运行在本地的集群环境中, 实现了应用及应用运行环境的自动化部署、伸缩及容错等, 使研究者不仅可以基于这个平台进行平台层云技术的研究和探索, 而且为云端应用和服务的研究提供了一个基础平台, 使得研究者可以利用本地的资源进行应用和服务创新。

12.7 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, 以及一些著名的开源云计算系统都进行了介绍。

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, 还提供了全方位的云计算解决方案, 其中包含了三个方面: 第一, 面向客户实际业务需求的开发测试云、桌面云、分析云、协作云和存储云等服务; 第二, 给客户提供了搭建私有云环境所必需的软件管理系统, 比如TSAM、System Director和WCA, 第三, 提供了全面的云计算业务咨询服务和技术咨询服务。

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和平台云服务集合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、提供计算资源的Amazon EC2服务，以及新推出的关系数据库服务RDS。

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、打包服务等。

Microsoft公司也凭借最近推出的Windows Azure platform云计算平台在云计算业界占有了一席之地。Windows Azure platform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 and 存储等资源。我们所介绍的平台层服务包含了与应用开发相关和运行相关的Windows Azure platform AppFabric、与数据管理有关的SQL Azure服务。随后还介绍了Microsoft在云应用层提供的Live服务，包括Windows Live和Office Live。

开源社区在推动软件发展过程中有不可泯灭的贡献，在最后一节我们对开源软件中典型的基础设施云方案Eucalyputs、Nimbus及平台层云方案AppScale进行了介绍。

希望通过这一章的内容，起到抛砖引玉的作用，为读者提供一个能够快速上手的参考，开启您的云计算之旅。



附录A: 参考文献

- [1] Wikipedia.com: Cloud computing, http://en.wikipedia.org/wiki/Cloud_computing
- [2] Searchcloudcomputing.com: What is cloud computing, http://searchcloudcomputing.techtarget.com/sDefinition/0,,sid201_gci1287881,00.html#
- [3] Microsoft: Cloud computing definition, http://www.microsoft.com/china/CRD/en/innoforum/innoforum_14.aspx
- [4] Salesforce: Cloud computing definition, <http://www.salesforce.com/cloudcomputing/>
- [5] BusinessWeek: Google and the Wisdom of Clouds, http://www.businessweek.com/magazine/content/07_52/b4064048925836.htm
- [6] BusinessWeek: Amazon and Cloud Computing, <http://www.eweek.com/c/a/Cloud-Computing/Amazon-and-Cloud-Computing>.
- [7] 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>.
- [8] Hagit Attiya and Jennifer Welch: Distributed Computing: Fundamentals, Simulations, and Advanced Topics. ISBN 0-471-45324-2, published by John Wiley and Sons, Inc
- [9] Utility computing, IBM System Journal, <http://www.research.ibm.com/journal/sj43-1.html>
- [10] IBM: SOA 和 Web Services 入门, IBM developerWorks中国<http://www.ibm.com/developerworks/cn/webservices/newto/index.html>
- [11] Chang Hua Sun, et al. Simplifying Service Deployment with Virtual Appliances, in Proceeding of International Conference on Service Computing, SCC' 2008, July 2008, Honolulu, Hawaii, USA, pages 265- 272.
- [12] 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>
- [13] 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/>
- [14] GigaOM's Refresh the net Report: Why the Internet need a makeover? , <http://www.scribd.com/doc/3569671/GigaOMs-Refresh-the-Net-Report>
- [15] Cloud computing Economies of scale, http://mvdirona.com/jrh/TalksAndPapers/JamesHamilton_SMDB2009.pdf.

- [16] Cloud Computing Types: Public Cloud, Hybrid Cloud, Private Cloud, http://www.circleid.com/posts/20090306_cloud_computing_types_public_hybrid_private/
- [17] PARKHILL, D. The Challenge of the Computer Utility. Addison-Wesley Educational Publishers Inc., US, 1966.
- [18] Amazon Web Services: Case Studies, <http://aws.amazon.com/solutions/case-studies/>
- [19] Amazon Web Services Blog: Success stories from Europe and Asia, <http://aws.typepad.com/aws/2009/03/european-success-stories.html>
- [20] InformationWeek: Amazon, Others Explore the Cloud for Medical Research, Health Care, <http://www.informationweek.com/news/services/storage/showArticle.jhtml?articleID=212201778>
- [21] Salesforce.com: <http://www.salesforce.com/cn/?ir=1>
- [22] Services Computing, Liang-Jie Zhang, Jia Zhang, Hong Cai,
- [23] 《历史的起源与目标》，第二篇，第一章，第二节<现代技术>，卡尔·雅斯贝斯
- [24] 中国移动发布大云1.0系统, <http://tech.163.com/10/0521/17/677M324B000915BE.html>
- [25] 中国电信e云存储, <http://www.51eyun.com/index.do>
- [26] 让科技引领中国可持续发展 http://www.gov.cn/ldhd/2009-11/23/content_1471208.htm
- [27] Amazon's EC2 Generating \$220M Annually, Randy Bias <http://cloud.kendallsquare.com/article/amazons-ec2-generating-220m-annually>
- [28] Google's CEO on the Power of Clouds , http://www.businessweek.com/magazine/content/07_52/b4064052938160_page_2.htm
- [29] IBM SAN Volume Controller, <http://www-03.ibm.com/systems/storage/software/virtualization/svc/index.html>
- [30] 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.
- [31] Hadoop Distributed File System, http://hadoop.apache.org/core/docs/current/hdfs_design.html
- [32] VMware Virtual Machine File System, <http://www.vmware.com/products/vi/esx/vmfs.html>
- [33] Neiger, Gil; A. Santoni, F. Leung, D. Rodgers, R. Uhlig. Intel Virtualization Technology: Hardware Support for Efficient Processor Virtualization, Intel Technology Journal(Intel)10(3), July 2008, pages: 167-178.
- [34] AMD Virtualization Technology, <http://www.amd.com/us/products/technologies/virtualization/Pages/amd-v.aspx>
- [35] Amazon CloudWatch Monitoring Model, <http://docs.amazonwebservices.com/AmazonCloudWatch/latest/DeveloperGuide/>

- [36] Salesforce.com Acknowledges Data Loss, http://voices.washingtonpost.com/securityfix/2007/11/salesforcecom_acknowledges_dat.html
- [37] Lei Shi, et al. Iceberg: An Image Streamer for Space and Time Efficient Provisioning of Virtual Machines. International Conference on Parallel Processing – Workshops, 2008, pages 31–38.
- [38] Brendan Cully, Geoffrey Lefebvre, Dutch Meyer, Mike Feeley, Norm Hutchinson, and Andrew Warfield (April 2008). Remus: high availability via asynchronous virtual machine replication. In Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, San Francisco, California, pages. 161–174. ISBN 111–999–5555–22–1.
- [39] Amazon EC2: <http://aws.amazon.com/ec2>
- [40] JDBC 4.0 Specification, <http://jcp.org/en/jsr/detail?id=221>
- [41] Ewan Mellor, et al: Xen Management API version 1.0.6
- [42] 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>
- [43] Google App Engine, <http://code.google.com/appengine/>
- [44] Java Enterprise Edition (JavaEE) Specification, J2EE规范, <http://jcp.org/aboutJava/communityprocess/final/jsr244/index.html>
- [45] Yefim V. Natis: Reference Architecture for Multitenancy: Enterprise Computing “in the Cloud”, Gartner. com publication, ID Number G00163395
- [46] Robert P. Desisto, Ben Pring: Essential SaaS Overview and 2009 Guide to SaaS Research, Gartner Research, ID Number: G00167279, 23 April 2009.
- [47] Yefim V. Natis. Reference Architecture for Multitenancy: Enterprise Computing “in the Cloud”, Gartner Research Publication, ID number: G00163395
- [48] Stefan Ried. Forrester’s SaaS Maturity Model, Transforming Vendor Strategy While Managing Customer Expectations. Forrester.com, April 14, 2008.
- [49] Lars-Olof Burchard, Matthias Hovestadt, et al. The Virtual Resource Manager: An Architecture for SLA-aware Resource Management, IEEE International Symposium on Cluster Computing and the Grid 2004, Pages 126 – 133, April 2004.
- [50] Karl Czajkowski, Ian Foster, et al. SNAP: A Protocol for Negotiating Service Level Agreements and Coordinating Resource Management in Distributed Systems. In 8th Workshop on Job Scheduling Strategies for Parallel Processing, pages 153–183, Springer Verlag Berlin Heidelberg 2002.
- [51] 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, pages 323-356.
- [52] Armando Fox, Steven D. Gribble, Yatin Chawathe, Eric A. Brewer, and Paul Gauthier. Cluster-based scalable network services. In Proceedings of the 16th ACM Symposium on Operating System Principles, Saint-Malo, France, 1997, pages 7-91.
- [53] Luiz A. Barroso, Jeffrey Dean, and Urs Holzle. Web search for a planet: The Google cluster architecture. *IEEE Micro*, April 2003, 23(2):22-28.
- [54] 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 Parallel and Distributed Systems (IOPADS '99), Atlanta, Georgia, May 1999, pages 10-22.
- [55] 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, pages. 137-150.
- [56] Apache Hadoop, <http://hadoop.apache.org/>
- [57] T. Johnson. Designing a distributed queue. In Proceeding of the 7th IEEE Symposium on Parallel and Distributed Processing, pages: 304-311.
- [58] Java Message Service (JMS) Specification, <http://jcp.org/en/jsr/d>
- [59] Apache ActiveMQ, <http://activemq.apache.org/>
- [60] IBM Websphere MQ, <http://www-01.ibm.com/software/integration/wmq/>
- [61] Ellard T. Roush, Roy H. Compbel. Fast Dynamic Process Migration. In proceedings of the 6th IEEE International Conference on Distributed Computing System, 1996, pages: 637-645.
- [62] Brendan Cully, Geoffrey Lefebvre, et al. Remus: High Availability via Asynchronous Virtual Machine Replication. In Proceedings of the 5th USENIX Symposium on Networked System Design and Implementation NSDI' 08, 2008, pages:161-174.
- [63] Chandramohan A. Thekkath, Timothy Mann and Edward K. Lee. Frangipani: A Scalable Distributed File System, In Proceedings of the 16th ACM Symposium on Operating Systems Principles, SOSP' 97, 1997, pages: 224-237.
- [64] Björn Grynwall, 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), pages: 251-264.
- [65] Sage A. Weil, Scott A. Brandt, Ethan L. Miller, Darrell D.E. Long and Carlos Maltzahn. Ceph: A Scalable,

- High-performance Distributed File System. In Proceedings of the 7th Symposium on Operating Systems Design and Implementation OSDI' 06, 2006, pages: 307–320.
- [66] S Ghemawat, H Gobiuff, 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, pages: 29–43.
- [67] F Chang, J Dean, S Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, R. E. Gruber. Bigtable: A Distributed Storage System for StructuredData. OSDI 2006.
- [68] G DeCandia, D Hastorun, M Jampani, G Kakulapati, A Lakshman, A Pilchin, S Sivasubramanian, P Vosshall, W Vogels. Dynamo: Amazon's Highly Available Key-value Store. SOSP 2007.
- [69] H Yang, A Dasdan, RL Hsiao, DS Parker. Map-Reduce-Merge: Simplified Relational Data Processing on Large Clusters. Proceedings of the 2007 ACM SIGMOD.
- [70] Yu Hui Wu, Zhi Le Zou, et al. SPA: A Comprehensive Framework for Hybrid Solution Provisioning. In Proceedings of the IEEE 7th International Conference on Web Services, ICWS 2009, July 6–10, 2009, Los Angeles, CA, USA.
- [71] Smart Business Storage Cloud, <http://www-935.ibm.com/services/us/index.wss/offering/its/a1031610>
- [72] Cloud Computing and Storage. <http://www-03.ibm.com/systems/storage/solutions/cloud/index.html>
- [73] Smart Analytics Cloud for System z, <http://www-03.ibm.com/systems/z/solutions/cloud/smart.html>
- [74] Smart Business Desktop, <http://www-935.ibm.com/services/us/index.wss/offerfamily/eus/a1031846>
- [75] Smart Business Development and Test Cloud, <http://www-935.ibm.com/services/us/index.wss/offering/middleware/a1030965>
- [76] IBM Smart Business Development & Test, <http://www-180.ibm.com/cloud/developer/dashboard>
- [77] IBM: Cloud Computing Overview, <http://www.ibm.com/ibm/cloud/>
- [78] IBM: Cloud Computing Services, <http://www.ibm.com/ibm/cloud/services/>
- [79] IBM: Cloud Computing Systems, <http://www.ibm.com/ibm/cloud/systems/>
- [80] IBM: Cloud Computing Consulting, <http://www.ibm.com/ibm/cloud/consulting/>
- [81] IBM: IBM Smart Business Development and Test on the IBM Cloud,
- [82] <http://www-180.ibm.com/cloud/enterprise/beta/dashboard>
- [83] Cloudbook: RC2, <http://www.cloudbook.net/the-cloud-book/private-cloud/ibm>
- [84] Lotus Live, <https://www.lotuslive.com/zh/>
- [85] Google App Engine Blog: Back to the Future for Data Storage, <http://googleappengine.blogspot.com/2009/02/back-to-future-for-data-storage.html>
- [86] Wikipedia: Optimistic concurrency control, [http://en.wikipedia.org/wiki/Optimistic_](http://en.wikipedia.org/wiki/Optimistic_concurrency_)

control

- [87] Google Code: Queries and Indexes, <http://code.google.com/intl/en/appengine/docs/python/datastore/queriesandindexes.html>
- [88] Google Code: Datastore Python API Overview, <http://code.google.com/intl/en/appengine/docs/python/datastore/overview.html>
- [89] Google Code: The Administration Console, <http://code.google.com/intl/en/appengine/docs/theadminconsole.html>
- [90] Salesforce: Force.com Platform, <http://www.salesforce.com/platform/what-is-it.jsp>
- [91] Force.com: Developer Core Resource Library, <http://wiki.developerforce.com/index.php/DeveloperCoreResources>
- [92] Force.com: An Overview of Force.com Security, http://wiki.developerforce.com/index.php/An_Overview_of_Force.com_Security
- [93] Force.com: Database Services, http://wiki.developerforce.com/index.php/Database_Services
- [94] Force.com: Create and Run any Application, On Demand, http://wiki.developerforce.com/index.php/Force.com:_Create_and_Run_any_Application,_On_Demand
- [95] Elastic Site, Using Clouds to Elastically Extend Site Resources, Paul Marshall, Department of Computer Science, University of Colorado at Boulder, Boulder, CO, USA
- [96] Infrastructure-as-a-Service(PPT), Cloud Computing for Science, October 2009, Banff Centre, Banff, Canada, Kate Keahey
- [97] Cloud Computing with Nimbus (PPT), XtremOS Summer School 2009, Oxford, September 2009, Kate Keahey
- [98] Sky Computing, Keahey, K., Tsugawa, M., Matsunaga, A., Fortes, J. IEEE Internet Computing, vol. 13, no. 5, September/October 2009.
- [99] 15 Elastic Site: Using Clouds to Elastically Extend Site Resources, Marshall, P., Keahey K., Freeman, T. IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2010), Melbourne, Australia. May 2010
- [100] M.H. Dodani, Cloud Redux, Journal of Object Technology, Vol. 8, No. 5: pp 31–38
- [101] M. Gillis, Cloud Computing—Expanding IT flexibility and agility, ICT Spring Event Europe Luxemburg, March 2010, Access: <http://www.ictspring.com/pdf/cr-d2-1045-marnix-gillis.pdf>
- [102] Cloud Security Alliance, Cloud security guidance for critical areas of focus in cloud computing v2.1, Dec. 2009.

- [103] IBM Global Business Services white paper, Business strategy for cloud providers - the case for potential cloud services providers, Sep. 2009.
- [104] IBM Global Business Services white paper, Capturing the potential of cloud-how cloud drives value in enterprise IT strategy, Sep. 2009.
- [105] Windows Azure Developer Center, <http://msdn.microsoft.com/en-us/windowsazure/default.aspx>
- [106] Microsoft Office Live Workspace, <http://workspace.office.live.com>
- [107] Microsoft Office 2010, <http://office.microsoft.com/zh-cn/>
- [108] Microsoft Windows Live, <http://www.windowslive.com>
- [109] Cloudbook: RC2, <http://www.cloudbook.net/private-cloud/ibm>
- [110] Google App Engine Blog: Back to the Future for Data Storage, <http://googleappengine.blogspot.com/2009/02/back-to-future-for-data-storage.html>
- [111] Online collaboration: LotusLive, <https://www.lotuslive.com/services>
- [112] Microsoft: Windows Azure Platform, <http://www.microsoft.com/azure/default.mspx>
- [113] David Chappell. Introducing the Azure Service Platform, May, 2009 <http://www.microsoft.com/presspass/events/pdc/docs/AzureServicesPlatform.pdf>
- [114] Wikipedia: Amazon.com, <http://en.wikipedia.org/wiki/Amazon.com>
- [115] Amazon web services: Amazon Simple Storage Service (Amazon S3), <http://aws.amazon.com/s3/>
- [116] Amazon Web Services: Amazon Simple Storage Service Getting Started Guide(API Version 2006-03-01), <http://docs.amazonwebservices.com/AmazonS3/latest/gsg/>
- [117] James Murty. Programming Amazon Web Services, O' REILY, 2008, ISBN 10: 0-596- 51581-2(ISBN 13: 9780596515812)
- [118] Amazon Web Services: Amazon SimpleDB, <http://aws.amazon.com/simpledb/>
- [119] Amazon Web Services: Amazon SimpleDB Getting Started Guide <http://docs.amazonwebservices.com/AmazonSimpleDB/latest/GettingStartedGuide/>
- [120] Amazon Web Services: Amazon Simple Queue Service (Amazon SQS), <http://aws.amazon.com/sqs/>
- [121] Amazon Web Services: Amazon Simple Queue Service Getting Started Guide (API Version), <http://docs.amazonwebservices.com/AWSSimpleQueueService/latest/SQSGettingStartedGuide/>
- [122] Amazon Web Services: Amazon Elastic Compute Cloud (Amazon EC2), <http://aws.amazon.com/ec2/>
- [123] Amazon Web Services: Amazon Elastic Compute Cloud Getting Started Guide (API Version), <http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/>
- [124] Eucalyptus Community: <http://open.eucalyptus.com/>

- [125] Appscale : <http://appscale.cs.ucsb.edu/>
- [126] Appscale : <http://code.google.com/p/appscale/>
- [127] Eucalyptus: A Technical Report on an Elastic Utility Computing Architecture Linking Your Programs to Useful Systems, UCSB Computer Science Technical Report Number 2008-10
- [128] Intel? Cloud Builder Guide to Cloud Design and Deployment on Intel? Platforms – Ubuntu Enterprise Cloud
- [129] The Eucalyptus Open-source Cloud-computing System, Daniel Nurmi, Rich Wolski, Chris Grzegorzczak, Graziano Obertelli, Sunil Soman, Lamia Youseff, Dmitrii Zagorodnov
- [130] AppScale Design and Implementation, Navraj Chohan Chris Bunch Sydney Pang Chandra Krintz Nagy Mostafa Sunil Soman Rich Wolski, Computer Science Department, University of California, Santa Barbara
- [131] AppScale: Scalable and Open AppEngine Application Development and Deployment
- [132] Google App Engine: <http://code.google.com/appengine/>
- [133] Wikipedia: Software Architecture, http://en.wikipedia.org/wiki/Software_architecture
- [134] Edsger Dijkstra. Go-to statement considered harmful, in Commun. ACM 11 (1968),3: 147-148.
- [135] Ali Arsanjani, et al. Design an SOA solution using a reference architecture. <http://www.ibm.com/developerworks/library/ar-archtemp/>
- [136] "YouTube hits 100m videos per day", BBC technology analysis report, available at <http://news.bbc.co.uk/1/hi/technology/5186618.stm>
- [137] IBM Cloud Computing, <http://www.ibm.com/cloud/>
- [138] Sanjay Ghemawat and etc; The Google File System; SOSP 2003
- [139] HDFS; <http://hadoop.apache.org/hdfs/>
- [140] Doug Beaver and etc; Finding a needle in Haystack: Facebook's photo storage; OSDI 2010
- [141] 浅议DAS、NAS、SAN三种模式; <http://www.it.com.cn/f/server/053/21/89080.htm>
- [142] 张冬; 大话存储:网络存储系统原理精解与最佳实践; 清华大学出版, ISBN: 9787302186724
- [143] Avinash Lakshman, Prashant Malik; Cassandra – A Decentralized Structured Storage System; LADIS 2009
- [144] Fay Chang, Jeffrey Dean and etc; Bigtable: A Distributed Storage System for Structured Data; OSDI 2006
- [145] Mike Stonebraker; Urban Myths about SQL; <http://voltdb.com/voltdb-webinar-sql-urban-myths>
- [146] Protocol Buffers; <http://code.google.com/intl/zh-CN/apis/protocolbuffers/docs/overview.htm>
- [147] Justin Sheehy; Riak Inside; Basho Technologies
- [148] Hbase; <http://hbase.apache.org/>

- [149] Andrew S. Tanenbaum; Distributed Systems: Principles and Paradigms (2nd Edition); Prentice Hall, ISBN-10: 0132392275
- [150] SaaS Architecture, the white paper, Progress Software, 2008
- [151] Core Saas Desig and Architecture Patterns, presentation from Duane Nickull
- [152] George Coulouris, Jean Dollimore and Tim Kindberg; Distributed Systems: Concepts and Design (4th Edition); Addison Wesley, ISBN-10: 0321263545
- [153] NoSQL; <http://www.yankay.com/>
- [154] Red Hat; <http://www.redhat.com/f/pdf/rhev/DOC-KVM.pdf>
- [155] Wiki; http://en.wikipedia.org/wiki/KVM_switch

