

Cloud Foundry

从数字化战略到实现

冯雷 高小明 吴疆 付宁 编著

Pivotal公司官方出品，企业实现数字化战略必读

提供从数字化转型战略规划到基于Cloud Foundry实现的整体解决方案



机械工业出版社
China Machine Press

云计算与虚拟化技术丛书

Cloud Foundry

从数字化战略到实现

冯雷 高小明 吴疆 付宁 编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Cloud Foundry: 从数字化战略到实现 / 冯雷等编著. —北京: 机械工业出版社, 2017.7
(云计算与虚拟化技术丛书)

ISBN 978-7-111-57571-9

I.C… II.冯… III.云计算 IV.TP393.027

中国版本图书馆 CIP 数据核字 (2017) 第 189386 号

本书从计算技术的发展及其对商业模式的影响谈起,介绍了云计算技术给现代社会和商业带来的变革,阐述了云计算,特别是 PaaS 云提出的必要性及其为企业数字化转型带来的机会,以及企业在数字化战略中得到的变革。在此基础上,介绍了 PaaS 平台 Cloud Foundry 的架构、部署、实施、运维,使企业顺利基于 Cloud Foundry 开发云原生应用,实施数字化转型。

本书适合希望通过云计算技术实现数字化转型的企业决策者、CIO、从事云计算技术工作和咨询工作的读者阅读,也适合希望了解 PaaS 云技术的读者和研究人员阅读。

Cloud Foundry: 从数字化战略到实现

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 朱 劼

责任校对: 李秋荣

印 刷: 北京市荣盛彩色印刷有限公司

版 次: 2017 年 9 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 16.5

书 号: ISBN 978-7-111-57571-9

定 价: 79.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东



IT 喜來 | Information Technology
HZBOOKS

当今，IT 行业正在经历一场前所未有的巨大变革。虽然 IT 技术本身的特点决定了这个行业注定充满变化，但在本人 28 年的 IT 从业经历中，还是第一次感受到像今天这样规模巨大且影响深远的变革。IT 技术日新月异的发展给各行各业带来了翻天覆地的变化，甚至可以说推动了整个人类社会从传统经济到数字经济的转型：从金融、零售、新闻传媒，到交通、通信、社交、媒体、政府政务，这些领域无一不受到 IT 技术发展所带来的巨大影响，面临着数字转型的挑战和机遇。数字经济所带来的变化之快、影响之大，令许多人始料未及。短短几年中，数字经济已经得到长足发展。今天，全球最大的短租服务企业 Airbnb 没有一间自己的实体酒店，全球最大的零售企业阿里巴巴没有自己的库房和店面，全球最大的出租车公司 Uber 没有一辆自己的出租车——支撑数字经济的，是数字技术（或者 IT 技术）。摒弃了传统经济的运营模式，这些核心技术所要解决的根本问题是：第一，随时随地的万物互联（移动互联网）；第二，无穷无尽的计算资源（云计算）；第三，极其丰富的数据智慧（大数据）。例如，前面提到的 Airbnb，从根本上来说，它解决了房屋提供者和需求者之间的联系问题，通过互联网和云计算技术构建了一个拥有成千上万卖家和买家的交易平台，这就是典型的数字经济，即买家和卖家无需见面，就能在虚拟数字平台上进行和完成所有交易。这个虚拟数字平台的前端是我们的手机或电脑，后台则是能够迅速处理海量交易请求和数据的云计算平台。

云计算从诞生到今天经历了持续的演进和变化，由于每个人看待云计算的角度不同，对云计算的定义也存在多种版本，多少有点盲人摸象的感觉。但随着各行各业，特别是消费领域开始大规模使用云计算平台来推动其行业转型，各种前端应用对后端云计算平台的要求也越来越清晰。无论云计算的标准定义是什么，今天我们所说的云计算至少应该具有以下几个特点：第一，伸缩性和扩展性，即系统必须能够自动满足各种负载需求。以 12306

火车售票系统为例，它既要能够处理平峰时的访问量，更要足以应对高峰时（如春运期间）1000 倍以上的访问量；第二，安全性，即系统本身一定要保证安全可用，毕竟云平台上存在各类应用和诸多用户，信息安全可谓重中之重；第三，灵活性，即云平台上的应用必须能够对用户需求的变化做出迅速响应。

要实现云计算系统的伸缩性、安全性和灵活性并非易事，特别是在动辄成千上万，甚至几十万台服务器的超大型云计算中心中实现上述功能。大家知道，要想让 IT 系统具有灵活性，就要从根本上把物理层面的操作变成逻辑层面的操作，把具体变为抽象，把由硬件完成的任务交由软件来完成。一个简单的类比是，当我们要物理移动一台笨重的机器时，必然会比在软件中移动 0 和 1 要费力费时。基于这样一个简单朴素的道理，我们就不难理解云计算中的那些关键技术，例如虚拟化、IaaS 及 PaaS。IaaS（Infrastructure as a Service）的功能从根本上来说是把大量物理的计算存储和网络资源虚拟化为虚拟资源池，用户通过软件调用这些资源，无需关心底层硬件，从而实现基础架构的灵活可伸缩性。同样的道理，为了让云计算应用的开发者无需关心基础架构的细节从而能够专注于应用本身的开发和部署，应用和基础架构之间的通信和管理就从具体应用中剥离出来，成为一个通用的软件平台，这个平台就叫做 PaaS（Platform as a Service）。就像 IaaS 用户不需要关心底层服务器一样，PaaS 平台用户也无需关心基础架构，只要关注于自己所要部署和开发的应用和服务即可。

目前市场上存在着许多 PaaS 产品，令人眼花缭乱，其中 Cloud Foundry 是出现最早、迄今为止在业界应用最为广泛的 PaaS 技术。Cloud Foundry 最早由 Pivotal 公司开发，为了推动云计算的发展，Pivotal 公司和 VMware 公司于 2015 年将此项技术开源，以吸引更多的用户和开发者来参与开发。本书的几位作者是亲自参与 Cloud Foundry 开发的非常优秀的工程技术人员，他们不仅对该项技术本身有着非常透彻的了解，同时还有丰富的应用 Cloud Foundry 技术的经验。计算机科学说到底是一门应用科学，由这样一批既深谙 PaaS 技术核心，又有着丰富实操经验的人士，根据其亲身经历和工作体会来编写这本关于 Cloud Foundry 的参考读物，可谓再合适不过了。相信无论是对云计算感兴趣的工程技术人员，还是正在寻求知识拓展的在校学生，都会发现此书是一本不可多得的参考书。

祝大家阅读愉快。

刘伟

EMC 公司全球副总裁，中国研发集团总经理

2017 年 1 月于北京

一、为何要编写这本书

Pivotal 中国研发中心在 2013 年伴随着 Pivotal 全球总公司创立而同步组建，中国研发中心的使命是为企业数字化变革提供前瞻性的技术支撑、方法和文化。Pivotal 中国研发中心团队继承自 EMC 卓越研发集团的大数据团队和 VMWare 研发中心的云分析（Cloud Analytics）团队。今天，中国研发中心主要围绕 Cloud Foundry 的 Web Service 服务、Greenplum 大数据产品和 Hadoop 上的 HAWQ 大数据引擎进行研发。这些产品组合在一起，为企业数字化转型提供了基于 PaaS 并拥抱大数据的现代敏捷基础架构。在云时代，非数字型企业采用这个基础架构可以极大地缩短和世界级互联网企业投入数亿美元研发的基础设施的差距。

Pivotal Cloud Foundry 是 Pivotal 产品套件的基础，中国 Cloud Foundry 团队秉承探索精神，在 2010 年就开始对云和数据交集的前沿进行探索。2010 年年底，我们尝试着想象 2015 年到 2020 年工业界云和数据的尖端并开始征程，中间也触发了 Cetas 的并购以加快征程。随着时间的推移，我们创造的产品和服务一次次被时代浪潮托起。团队今天运营着 Cloud Foundry 的云版本服务，迎接 PaaS 浪潮的到来。

因为 Pivotal 公司秉承开源精神，所以我们的这些产品集可以用于企业的数字化转型。但是，在将 PaaS 云和大数据套件平民化的过程中，非数字型企业面临战略定位、技术培养和文化转型的挑战。这也触发了我们研发中心的同事们编写著作的想法，希望能够帮助实施转型的企业定义战略、开发云原生应用以提供数字化服务，并提供 Cloud Foundry 运营方面的帮助。

二、为何要立足 PaaS

在过去很长的时间里，系统集成商在企业服务中扮演了很重要的角色。他们帮助企业选择硬件、网络、软件并且配置、部署，原因在于企业软件往往很复杂，而这个复杂性从部署安装就开始存在了，企业不得不通过系统集成商来完成这些工作。另一方面，缺乏经验的工程师可能要摸索比较长的时间才能让产品运行在稳定可靠的状态。网络工程师和系统工程师往往忙得不可开交，经常要跑到机房去连接网络和安装不同的系统，产品出现问题时总要从软件到硬件逐层排查，费时费力。

IaaS (Infrastructure as a Service) 技术的成熟让大家轻松了很多，所有的网络、存储和计算资源都可以用一个集中的平台加以有效的管理，需要使用时创建一个按需配置的虚拟机即可。可以说，IaaS 把硬件资源抽象化了。IaaS 的好处显而易见，而且应用广泛，所以大家在说到云计算的时候，有时候会把 IaaS 看成云计算的全部。其实，在云计算诞生之初，人们就设想了在 IaaS 之上建设一个 PaaS (Platform as a Service) 层。IaaS 层提供虚拟计算、网络和存储资源之后，部署应用时需要的运行时和服务还是很费周折，应用的生命周期管理、日志、监控都有繁重的工作。PaaS 就是为核心业务逻辑的应用提供支撑的服务平台，它抽象了应用之下、操作系统之上的所有软件资源和服务，让它们对应用友好而便捷。

PaaS 带来的价值一点都不亚于 IaaS。安装单节点的数据库很简单，安装多节点的数据库就复杂一点；运维一个多节点的数据库并保证很高的可用性，要求就更高了；如果再要求这个数据库的性能非常好，那么没有丰富的经验和完备的监控，就只能是一个美好的愿望。如果用户有成百上千个应用，而这些应用又依赖无数错综复杂的服务，那么 PaaS 平台正是这种场景的用武之地，它可以高效地管理好这些应用和服务，把所有这些应用和服务的公共需求做到极致好用。PaaS 让弹性计算真正得以实现。在没有 PaaS 时，把应用从 1 个节点扩展到 10 个节点，或者从 10 个节点减少到 1 个节点，需要做很多的工作，包括逐个创建虚拟机、逐个配置、逐个监控。有了 PaaS 之后，只需要一个命令，PaaS 就会自动执行以上所有过程。

简单地说，IaaS 是面向硬件资源的抽象，解放了系统工程师；PaaS 则提供了对软件服务的抽象，是应用开发和运维人员的得力助手。它们的共同点是，实现了共享资源和集中管理，降低了维护和操作成本，使效益和效率最大化。

我们相信，Cloud Foundry 会是一个对产业界有深远影响力的开源 PaaS 产品。经过多

年的技术积累和产品演化，Cloud Foundry 已经茁壮成长为一个可以部署在企业生产环境的成熟产品。在这样一个软件无处不在的时代，新型的互联网公司因为技术的积累和人才的优势，正在不断融合到市场的各个方向。传统的以业务为重的公司因为缺乏快速搭建丰富而可靠的应用的能力，在竞争中处于劣势。Cloud Foundry 的成熟恰逢其时，能够为所有软件公司或者希望以软件带动业务的传统公司提供新平台，从而帮助企业完美转型，通过敏捷开发，紧随市场变化，不断拓展业务。

三、这本书写了什么

在每个时代，都有一些伟大的技术和产品影响整个产业界，每种技术的孵化、发展、演进和流行都离不开技术领袖前瞻性的创新研发，也离不开技术布道者不遗余力的宣传推广。Cloud Foundry 的奠基者和开发者多年来倾尽全力打造这款产品，我们也愿意为 Cloud Foundry 的推广出一份力，希望抛砖引玉，帮助国内更多企业更好地通过 Cloud Foundry 获得商业上的成功。

坦白地说，Cloud Foundry 看似很复杂，但如果您能静下心来读一读本书，就会发现它本身的设计是很精练而自然的，它在很认真地解决一个广泛而复杂的问题。如果您能按照本书的介绍进行实战，即使是在笔记本电脑上安装一个单机的 Cloud Foundry，执行一些操作，也会发现 Cloud Foundry 对用户很友好，操作很简单。

本书分为以下几部分。第 1 章和第 2 章分享了作者对云计算技术的理解、思考和对云计算发展趋势的判断，并以汽车行业为例，展现了云计算在助力传统企业数字化转型中的作用，以及如何通过 PaaS 实施企业的数字化转型。第 3 章和第 4 章介绍了 Cloud Foundry 的发展历史、架构、特色，以及如何基于 Cloud Foundry 开发云原生的应用。第 5 章介绍了 Cloud Foundry 的部署工具 BOSH。第 6 章和第 7 章是 Cloud Foundry 的实战章节，介绍了 Cloud Foundry 的规划、部署和实施。最后两章介绍了 Cloud Foundry 的服务和运维工具，对于拓展 Cloud Foundry 的可用服务和提升运维的自动化程度很有帮助。

我们希望这本书能给准备借助云计算技术向数字化转型的企业决策者、CIO、正在从事云计算技术工作的读者带来帮助。作为学习型企业和团队，我们更建议采用如下方式阅读本书：企业的决策者阅读第 1 章和第 2 章，理解云战略（特别是 PaaS 云）提出的必要性和带来的数字化转型机会；之后将书传给软件开发团队，建议团队技术人员阅读第 3 章和第 4

章，这部分内容结合实际例子解释了 Cloud Foundry 的机理和云原生的概念；开发团队启动 Cloud Foundry 上的应用开发以后，可以把书传给云运维团队，他们可以重点阅读第 5 ~ 9 章，这部分内容会帮助他们理解如何在私有云或者公有云上运维 Cloud Foundry。

目前，已有不少传统企业基于 Cloud Foundry 进行数字化转型，更有很多创业团队基于 Cloud Foundry 开展他们的工作。关于如何创建一个适合数字化经济的新型创新软件和数据团队，作者非常想听到读者的问题和经验。您可以关注本书微博号（@Cloud Foundry: 从数字战略到实现）并留言。

限于作者的学识，本书难免有疏漏之处，恳请同行和各位读者批评指正，我们将不胜感激。

冯雷 (Ray Feng)

Pivotal 中国研发中心总经理

高小明 (Simon Gao)

Pivotal Cloud Foundry 高级研发经理

Acknowledgements 致谢

我们特别感谢为 Cloud Foundry 产品持续投资的 Pivotal 公司，感谢在持续不断改进 Cloud Foundry 的研发团队。

Pivotal 研发中心的前身 EMC Greenplum 研究部门在范承工博士和 Greenplum 创始人 Scott Yara 的关心下成立，这里一并感谢。也感谢 EMC 中国研究院院长刘伟博士和 VMware 中国研发中心总经理 Alan Ren 对 Pivotal 中国研发中心和 Cloud Foundry 的一贯支持。他们的帮助是本书能够出版的前提。

同时也感谢为本书出版提供编辑、审稿和出版的机械工业出版社华章分社。

感谢 Pivotal 研发中心执行助理段旻 (Ariel Duan) 为本书出版所作出的项目管理。

感谢 Pivotal Cloudops 中国团队所有成员对本书的贡献，特别是朱磊对第 7 章和第 9 章部分内容的整理、编写以及赵园对第 5 章内容的整理、编写。

感谢 Pivotal 的 Scott Frederick，本书对于应用开发的讲解基于 Scott 的演示项目。

最后，感谢所有为本书提供支持鼓励、内容建议和书稿审阅的朋友，这里不再一一列举。

目 录 Contents

序	
前 言	
致 谢	
第1章 数字化变革下的新云浪潮	1
1.1 计算技术的变迁与发展	1
1.1.1 计算技术经历的三代平台及其背后的技术公司	2
1.1.2 三代技术平台所驱动的商业模式	5
1.2 云计算概述	8
1.2.1 I层云	9
1.2.2 S层云	12
1.2.3 P层云	13
1.3 P层云计算和数字化变革	15
第2章 数字化转型第一步：选择PaaS云平台	18
2.1 数字化转型带来的新需求	19
2.1.1 车联网	20
2.1.2 运输网络和行程共享	20
2.1.3 无人驾驶	21
2.1.4 智能维护	22
2.1.5 电动汽车	22
2.2 数字化转型中的精益方法	22
2.3 数字平台选型	23
2.4 组建软件开发团队	25
第3章 Cloud Foundry：开源的PaaS云平台	29
3.1 Cloud Foundry 的历史	30
3.2 Cloud Foundry 的架构	31
3.3 应用准备和运行	36
3.4 Cloud Foundry 的特色	38
3.4.1 微服务	38
3.4.2 可伸缩	40
3.4.3 运维友好	41
3.4.4 高可用	43
3.4.5 可扩展	47
3.4.6 容器化	50
3.4.7 安全	51
3.5 Cloud Foundry 的应用案例	52
3.6 本章小结	53

第4章 基于Cloud Foundry开发

云原生的应用 55

- 4.1 云原生 55
- 4.2 持续发布和 Concourse 57
- 4.3 微服务和 Spring Boot 58
- 4.4 企业云原生应用的开发 59
 - 4.4.1 应用案例描述 59
 - 4.4.2 准备工作 60
 - 4.4.3 创建后端微服务 61
 - 4.4.4 前端子项目的开发 78
- 4.5 应用的管理 79
- 4.6 更多 Spring Cloud 服务 81
- 4.7 本章小结 85

第5章 BOSH原理与分析 86

- 5.1 BOSH 概述 86
 - 5.1.1 BOSH 的组件 87
 - 5.1.2 BOSH 部署的基本元素 89
- 5.2 安装与使用 BOSH 90
 - 5.2.1 使用 bosh init 安装
BOSH 90
 - 5.2.2 使用 bosh init 创建一个基于
AWS 的 BOSH 91
 - 5.2.3 BOSH 的用户设置 101
 - 5.2.4 使用 BOSH 106
 - 5.2.5 BOSH 命令详解 107
 - 5.2.6 BOSH 部署描述文件的
属性 111
 - 5.2.7 使用 BOSH 创建自己的
release 118

5.2.8 BOSH 中的 Health

Monitor 131

5.3 BOSH 故障排除 141

5.4 本章小结 141

第6章 Cloud Foundry的部署

规划 143

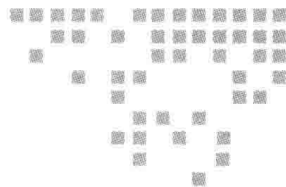
- 6.1 容量规划 143
 - 6.1.1 容量规划的概念和目标 143
 - 6.1.2 Cloud Foundry 的容量
规划 144
 - 6.1.3 Cloud Foundry 的容量
监控 145
- 6.2 网络设计 146
 - 6.2.1 网络设计的目标 146
 - 6.2.2 典型的网络设计方案 147
- 6.3 安全策略 149
 - 6.3.1 通过网络规划保证系统
安全 149
 - 6.3.2 Cloud Foundry 应用 (APP)
的安全 150
 - 6.3.3 应用 (APP) 级防火墙:
网络流量规则 151
 - 6.3.4 BOSH 的安全 151
 - 6.3.5 认证与授权 151
 - 6.3.6 安全事件的记录和审查 151
 - 6.3.7 安全性的一些建议 152
- 6.4 本章小结 152

第7章 Cloud Foundry的部署

实施 153

- 7.1 基于 AWS 的安装 153

7.1.1 安装的先决条件	153	8.2.3 Cloud Foundry Loggregator	214
7.1.2 部署 Cloud Foundry	161	8.3 度量收集与监控面板	216
7.2 基于 OpenStack 的安装	167	8.3.1 Datadog	216
7.2.1 安装的先决条件	167	8.3.2 OpenTSDB	217
7.2.2 部署 Cloud Foundry	172	8.4 系统监控与报警处理	218
7.3 基于 vSphere 的安装	182	8.5 日志的收集和分析	219
7.3.1 安装的先决条件	182		
7.3.2 部署 Micro BOSH	183	第9章 服务与工具	222
7.3.3 部署 Cloud Foundry	185	9.1 服务	222
7.4 使用 Ops Manager 安装	189	9.1.1 服务简介	222
7.4.1 OpenStack 支持版本	189	9.1.2 实现服务代理	225
7.4.2 安装的先决条件	190	9.1.3 管理服务代理	226
7.4.3 配置 OpenStack 环境	191	9.1.4 使用服务	228
7.4.4 配置并安装 Ops Manager Director	198	9.1.5 用户提供的服务	229
7.4.5 安装 Cloud Foundry	205	9.1.6 服务示例	230
7.5 本章小结	211	9.2 工具	235
第8章 Cloud Foundry的运维	212	9.2.1 PCF-Dev	235
8.1 度量	212	9.2.2 Concourse	240
8.2 度量的发送	213	9.2.3 Spiff	241
8.2.1 Bosh Health Monitor	214	9.3 本章小结	250
8.2.2 Cloud Foundry Collector	214	参考文献	251



数字化变革下的新云浪潮

作为云计算最早的一批从业人员，作者经历了云计算从兴起到蓬勃发展的历程。在云计算日益成为信息社会基础设施的今天，回顾并展望云计算的发展对于每个云计算从业者将具有重要的意义。本章中，作者将首先回顾包括云计算在内的三代技术平台和对应的数字商业模式，帮助读者从历史中发现变革模式和机遇。接下来聚焦到云计算，对云计算技术进行剖析，结合产品战略论证 P 层云（Platform as a Service, PaaS）为何将成为云计算下一阶段的发展热点。最后将和读者探讨云计算和数字化浪潮互为左右带来的社会变革和机遇。

为了帮助非技术专业读者理解云计算，本章会尽量采用通俗的语言，站在云计算产品领导层的视角从“三万英尺的高空”上总结云计算的技术概况。对于技术人员，我们将在后面的章节描述 P 层云计算部署的实战细节。

1.1 计算技术的变迁与发展

从 20 世纪发明计算机以来，我们的社会开始进入数字化时代。数字化之后的信息变成数据，以计算机可以处理的方式存储下来。信息技术的发展带来了数据量的不断增长，特别是最近几年，数据量呈爆发式增长。EMC 公司曾在 2014 年联合世界知名 IT 咨询公司 IDC 做了一个主题为《数字宇宙》的调查。该调查报告指出，2013 年全世界的数据量约为 4.4ZB，而到 2020 年，数据量将增长到 44ZB（如图 1-1 所示）。数字宇宙中，数据量基本每两年翻一番，这意味着近两年内创建的数据比之前积累的所有数据量还要大。

数字世界的不断膨胀对于计算机系统提出了新的要求，计算技术也因此得到不断发展。

简单来说，从计算机诞生到现在，计算技术经历了三代平台，每一代平台进化都发生了突破性的创新并带来商业模式的巨大变革。

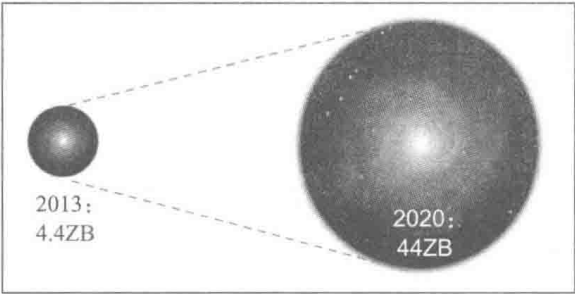


图 1-1 数字宇宙的不断膨胀

1.1.1 计算技术经历的三代平台及其背后的技术公司

从计算机诞生到今天，我们通常认为数字技术经历了如图 1-2 所示三代平台：以大型机（Mainframe）为代表的第一代平台，以小型机（Mini 或者 Minicomputer）和 PC（Personal Computer，也叫微型机）为代表的第二代平台和以云计算为基础的第三代平台。

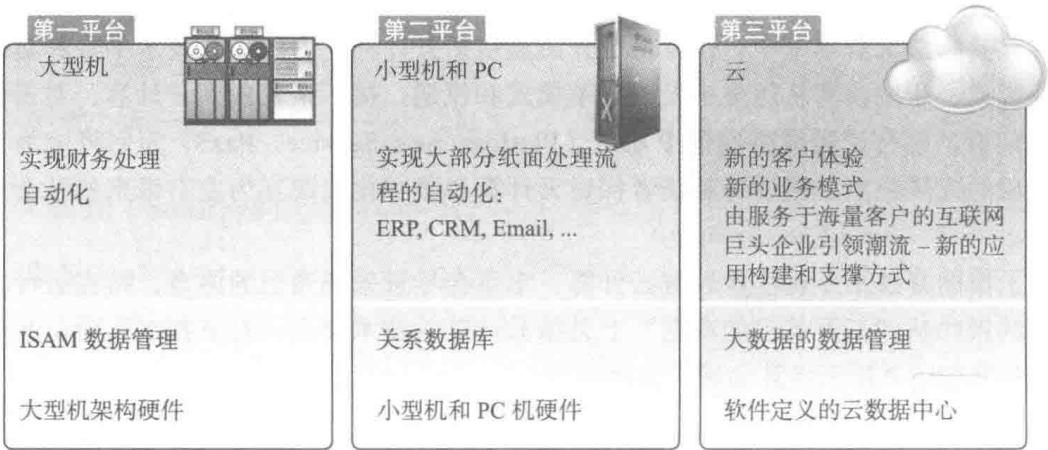


图 1-2 计算技术经历的三代平台

作者在 20 世纪 90 年代末进入北京大学时正好赶上 PC 浪潮，见证了 PC 时代的一大批公司的崛起和迅猛发展，包括个人机和企业服务器的生产厂商——惠普（HP）、康柏（Compaq）、戴尔（Dell）和蓝色巨人 IBM，当时最为人熟知、规模比较大的是微软公司和太阳微系统（Sun Microsystems）公司。现在大家熟悉的苹果公司虽然已经有了 Machintosh 机，但是在当时，加载微软 Windows 系统的 PC 在国内占绝对主导地位。当时主流的操作系统为 DOS、Windows 的各个版本（3.1, Windows 95 和 Windows NT）、UNIX 及其各种变体。最流行的 UNIX 变体是太阳微系统公司的 Solaris 操作系统。作者有幸体验过安装 Solaris 系

统的工作站，在上面写 Java 程序是当时很时髦的工作。UNIX 等同的开放源代码的操作系统 Linux 在 20 世纪 90 年代兴起，最终取代了 UNIX。

当时的数据管理系统主要建立在关系型数据库上，因此关系型数据库市场的竞争也相当激烈，包括大家至今仍非常熟悉的甲骨文（Oracle）、IBM DB2、微软的 SQL Server 和 Sybase 系统（已经被 SAP 收购）。

为了实现这些机器互联，逐渐出现了一批网络设备公司，从局域网快速扩展到互联网，其中比较知名的有 Novell 和思科（Cisco）公司。

在 PC 盛行之前，小型机占据主流地位。尽管名为小型机，但与今天的微型机相比，其外形还是要大很多。小型机的代表公司有 DEC、IBM、Data General、Wang 和 Prime。值得一提的是，华人王安（Wang An）创建的 Wang 计算机曾风靡一时。作为杰出华人代表，王安本人也曾经受到时任中央领导人邓小平的接见。图 1-3 为王安和他的 Wang 计算机。



图 1-3 王安和 Wang 计算机

王安祖籍江苏昆山，1920 年出生于上海，1940 年毕业于上海交大。1945 年留学于哈佛大学，1948 年获得应用物理博士学位。他在 1951 年创建了以他名字命名的实验室 Wang Laboratories，实验室生产的 Wang 计算器和 Mini 计算机 Wang2200 在市场取得巨大成功，曾经一度对蓝色巨人 IBM 构成威胁。

由于 PC 的很多设计思想和哲学继承自 Mini，UNIX 操作系统最早也出现在 Mini 上，因此这里把 Mini 和 PC 同时归属为第二代平台。在第二代平台及公司繁荣之前，同样存在过第一代平台和对应的伟大公司。作者没有经历过第一代平台，但是通过和公司的前辈交流知道，20 世纪 60 年代和 70 年代是大型机（Mainframe）盛行的时期，几大代表公司被称为 BUNCH，即 Burroughs、Univac、NCR、Control Data 和 Honeywell 的首字母缩写。与 Mini 和微型机的分而治之的设计思想不同，大型机强调单机硬件的处理能力。大型机年代，关系型数据思想尚未出现，IBM 的大型机采用了 ISAM（Index Searching Access Method）。IBM 虽然最先发布了关系型数据库理论，但最早开发关系型数据库的却是甲骨文公司。在第一代平台如日中天的时候，以 Mini 和 PC 代表的第二代平台开始悄悄发展。除了 IBM，其他第一代平台厂商没能把握这一趋势，最终没能成功过渡到第二代平台。即使在第二代平台迅速发展，PC 崛起的时候，大部分 Mini 计算机产商依然没有从 Mini 过渡到 PC 时代。

我们把思路从上个世纪拉回到现在，我们正处于第二代平台向第三代平台迁移的过程中，两代平台的公司并存着。第三代平台公司从一开始就建立在云计算基础上，他们的计算资源可以随着用户的增长而动态增长，他们的软件在用户不知不觉的情况下可以连续地发布升级，不会因系统升级导致服务暂停。用户可以通过多个终端（浏览器和手机