

微软技术开发者丛书

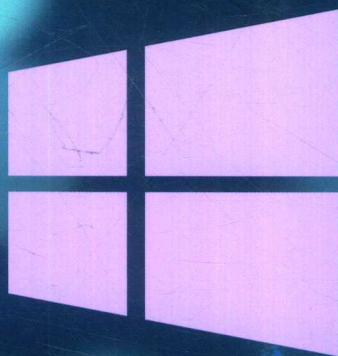
微软(中国)有限公司开发者体验与平台合作事业部策划出版

微软全球执行副总裁沈向洋作序

微软(中国)有限公司开发者体验与平台合作事业部总经理司瑞凯作序

# 微软开源跨平台移动 开发实践

利用 ASP.NET Core 1.0、Apache Cordova、Xamarin 和 Azure  
快速构建移动应用解决方案



Practicing Microsoft Open Source and Cross-Platform Mobile Development  
Building your Mobile Solution by ASP.NET Core 1.0, Apache Cordova, Xamarin and Azure

李争 张广昀 著

Li Zheng Zhang Guangyun

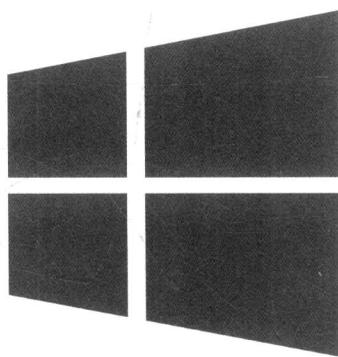
清华大学出版社



微软技术开发者丛书

# 微软开源跨平台移动 开发实践

利用 ASP.NET Core 1.0、Apache Cordova、Xamarin 和 Azure  
快速构建移动应用解决方案



Practicing Microsoft Open Source and Cross-Platform Mobile Development  
Building your Mobile Solution by ASP.NET Core 1.0, Apache Cordova, Xamarin and Azure

李 争 张广昀 著

Li Zheng Zhang Guangyun

清华大学出版社

北京

## 内 容 简 介

本书系统介绍了微软新一代高效、跨平台的应用平台 .NET Core 1.0。借助 .NET Core, 原本只能运行在 Windows 上的 .NET 应用可以运行在 Linux、OS X 甚至是 UNIX 上。结合微软的 Apache Cordova 和 Xamarin 等工具, 可以构建从服务端到客户端的完整开源、跨平台的解决方案。本书主要内容包括微软“云+端”战略, 革命性的 ASP.NET Core, 开源跨平台的设备端开发, 项目介绍和需求分析, 构建开发环境, 构建实体模型, 创建 Web 网站, 创建 Web API, 在 Azure 上部署服务, AngularJS 和 Ionic Framework, 使用 Cordova 开发跨平台移动应用, 使用 Xamarin 开发跨平台移动应用等内容。

本书详细论述了利用微软技术快速构建开源、跨平台的移动应用解决方案。本书并不限于仅仅介绍 .NET Core 技术, 而是向读者介绍如何将微软提供的一系列平台和技术串联起来, 形成确切方案进而为用户解决实际问题。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

### 图书在版编目(CIP)数据

微软开源跨平台移动开发实践: 利用 ASP.NET Core 1.0、Apache Cordova、Xamarin 和 Azure 快速构建移动应用解决方案/李争等著. —北京: 清华大学出版社, 2017

(微软技术开发者丛书)

ISBN 978-7-302-45360-4

I. ①微… II. ①李… III. ①移动终端—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2016)第 260831 号

责任编辑: 盛东亮

封面设计: 范华明

责任校对: 梁毅

责任印制: 杨艳

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社总机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈: 010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者: 北京密云胶印厂

经 销: 全国新华书店

开 本: 186mm×240mm

印 张: 14.75

字 数: 361 千字

版 次: 2017 年 1 月第 1 版

印 次: 2017 年 1 月第 1 次印刷

印 数: 1~2500

定 价: 59.00 元

产品编号: 072340-01

# 丛书序

FOREWORD

## 四十不惑,创新不止

从飞鸽传书到指尖沟通,从钻木取火到核能发电,从日行千里到探索太空……曾经遥不可及的梦想如今已经变为现实,有些甚至超出了人们的想象,而所有这一切都离不开科技创新的力量。

对于微软而言,创新是我们的灵魂,是我们矢志不渝的信仰。不断变革的操作系统,日益完善的办公软件,预见未来的领先科技……40年来,在创新精神的指引下,我们取得了辉煌的成绩,引领了高科技领域的突破性发展。

IT行业不墨守成规,只尊重创新。过往的成就不能代表未来的成功,我们将继续砥砺前行。如果说,以往诸如个人电脑、平板电脑、手机和可穿戴设备的发明大都是可见的;那么,在我看来,未来的创新和突破将会是无形的。“隐形计算”就是微软的下一个大事件。让计算归于“无形”,让技术服务于生活,是微软现在及未来的重要研发方向之一。

当计算来到云端后,便隐于无形,能力却变得更加强大;当机器学习足够先进,人们在尽享科技带来的便利的同时却觉察不到计算过程的存在;当我们只需通过声音、手势就可以与周边环境进行交互,计算机也将从人们的视线中消失。正如著名科幻作家亚瑟·查尔斯·克拉克所说:“真正先进的技术,看上去都与魔法无异。”

技术是通往未来的钥匙,要实现“隐形计算”,人工智能技术在这其中起着关键作用。近几年,得益于大数据、云计算、精准算法、深度学习等技术取得的进展,人工智能研究已经发展到现在的感知甚至认知阶段。未来,要实现真正的人机互动、个性化的情感沟通,计算机视觉、语音识别、自然语言将是人工智能领域进一步发展的突破口及热门的研究方向。2015年7月发布的Windows 10是微软在创新路上写下的完美注脚。作为史上第一个真正意义上跨设备的统一平台,Windows 10为用户带来了无缝衔接的使用体验,而智能人工助理Cortana、Windows Hello生物识别技术的加入,让人机交互进入了一个新层次。Windows 10也是历史上最好的Windows,最有中国印记的Windows,不但有针对中国本土的大量优

化,还会有海量的中国应用。Windows 10 是一个具有里程碑意义的跨时代产品,更是微软崇尚创新的具体体现,这种精神渗透在每一位微软员工的血液之中,激励着我们“予力全球每一人、每一组织成就不凡”。

四十不惑的微软对前方的创新之路看得更加清晰,走得也更加坚定。希望这套丛书不仅成为新时代之下微软前行的见证,也能够助中国的开发者一臂之力,共同繁荣我们的生态系统,绽放更多精彩的应用,成就属于自己的不凡。

沈向洋

微软全球执行副总裁

# 推荐序

FOREWORD

世上唯一不变的就是变化。对于高科技行业来说,自半导体和晶体管的发明以来,变化随时随地在发生。在 2016 年我们为变化赋予了新的名称——转型(Transformation)! 驱动转型的部分原因是 IT 基础设施投资的必然性;采用最新最强大的云计算基础结构的必然性——替代在原生代码中重新创造。这种现象可称为互联网 2.0/3.0/x.0。这种趋势非常明显——从传统的软件、传统的面向企业的服务转型为面向云、软件即服务的方式;从桌面驱动转型为处处驱动;从编写庞大的拥有知识产权的代码,转型为利用并贡献于开放源代码。

这种转型处处发生,而且不会遗漏包括消费者在内的任何一方,当然更不会遗漏任何一个行业。试想,当把传统的低技术含量的贩卖机转型成为可提供我们所述的精巧的云服务时,它就可以通过生物特征识别技术识别出当前使用者的情况,为之展示喜爱的商品,并根据用户选择的支付方式提供相应的服务:从传统的投币方式到包括 ApplePay、支付宝、微信支付在内的移动支付,应有尽有。然而,用户并不知道在背后支撑这些服务的是强大的云基础结构。除此之外,在这些技术发展的路径上,我们已经将汽车和由云技术及其发展的分支技术驱动的无人驾驶汽车,同高安全性、可扩展性和可用性联系起来。然而,这些也只是刚刚涉及这个领域的表层……我们正在开创增强/虚拟/混合现实世界的新时代。

另外一个无处不在的时髦概念是物联网(Internet of Things, IoT)。它涉及的场景是将软件植入物理世界的任何事物中,并根据场景要求来连接和应用——小型制造、环境工程、公共卫生和安全、医疗诊断和护理……这显然是通过互联网进行计算机之间对话的一场革命。但是,这个新的物联网世界提出的对计算能力、容量和安全的 yêu求,却是空前的。机器学习、物联网结构、生物特征识别的认知服务这些热门技术,对计算、存储的能力需求是空前的,而这种海量需求的存储和计算正是云计算平台所擅长的。我们相信微软 Azure 能够为 NASA、劳斯莱斯、政府机构、医疗诊断,以及在每个垂直行业提供无处不在的、超越想象的服务。

我的同事李争将通过本书引领读者领略全球最强大的、超大规模、高可用性和全能的云计算基础机构——微软的 Azure。书中还将涉及构建强大的计算架构以进行转型的行业方案,从利用开放源代码到跨平台技术和快速构建复杂的应用程序将企业的运维工作转型成为利用云服务,跨平台部署解决方案,学习和利用开放源代码技术等。书中详细介绍了 .NET Core 开源开发技术、Xamarin 移动跨平台开发技术,以及 Azure 云计算平台的开发

技术。

这个世界变化如此之快,各行各业都在转型,无论是医疗健康、加工制造,还是公共安全。我真诚地期望读者能够通过阅读此书,来构建出下一个杀手级的解决方案来改变这些现状。我非常肯定微软 Azure 一定能够帮你实现这个目标。祝你享受这个转型的过程,正如未来享受转型的结果一样。

司瑞凯(Srikanth Raju)

微软(中国)有限公司开发者体验与平台合作事业部总经理

# 前言

## PREFACE

2016年5月11日,我在新浪微博上留言“不是我不明白,这世界变化快啊!坐在微软的办公室里研究了一宿 Ubuntu Linux,五年前想都不敢想”,并配发了一张 Ubuntu gcc 编译开源项目时的截屏。是啊,这个世界变化真是太快了!别说五年前,就是三年前,一直从事微软技术路线的我也没有想过未来将要在 Linux 上部署我的应用。

微软公司在新任 CEO 的带领下开始踏上了新的征程,迅速地开始向开源领域转型。公司向开源领域转型的第一个动作就令人瞩目,微软首席执行官 Satya Nadella 在 2014 年 10 月直接对媒体喊出了“Microsoft Loves Linux”!紧接着,微软在 2014 年年底宣布 .NET Core 要支持跨平台,操作系统囊括了 Linux、OS X 甚至 FreeBSD,并且将 .NET Core 的代码放到了 Github 上。然后,微软又推出了广受好评的轻量级跨平台的 IDE: Visual Studio Code。至今开源界已经给 Visual Studio Code 贡献了上千个扩展插件。进一步,微软宣布 SQL Server 将在 2017 年正式支持 Linux,并且通过收购 Xamarin 公司一举揽收了 Gnome 和 Mono 项目的创始人 Miguel de Icaza 这种开源界明星级别的人物进入公司……

微软公司目前已经和开源界展开了紧密的合作,从云端大数据平台到移动端开发,在微软的产品中你都可以看到世界著名开源软件基金会 Apache 的身影。现在再讨论“要不要开源?要不要跨平台?”已经毫无意义了。微软的工程师现在只关心一个项目或者产品的哪部分要开源?要支持几种操作系统平台?

为什么要写这样一本书?这是一个好问题!经过这三年微软向开源和跨平台领域的转变,微软已经在开源和跨平台领域构建出了一套自己的技术体系。从服务器端的 .NET Core、ASP.NET 和 Web API,到 Web 端的 TypeScript 脚本语言,再到移动端的 Apache Cordova 和 Xamarin,微软已经可以支持开发者使用完全开源和跨平台的技术构建自己的应用。再配合免费的 Visual Studio Community 版本以及支持 Git 版本控制的 Visual Studio Online,开发者可以不花一分钱就可以在微软的平台上实现基于开源项目的应用程序生命周期管理。在李克强总理提倡的“大众创业,万众创新”的大潮中,微软现在的开源技术体系架构可以帮助创业者和开发人员大大降低创业的门槛和成本,迅速地将自己的创新想法变成现实。

作者编写这本书,主要是为了通过一个非常简单的电商项目,将目前微软开源跨平台的技术体系结构串联起来,帮助创业者和开发人员了解和掌握如何使用微软的开源跨平台技术来迅速构建自己的应用。然后启用微软云 Azure 的力量,快速部署和集成,迅速地将自



己的技术产品和想法推广到每个用户手中。

本书不是一本单独讲解某个微软开源技术的图书。恰恰相反,本书是把微软开发工具和开发服务(Visual Studio 2015、Visual Studio Online),微软开源开发技术(.NET Core 1.0、Apache Cordova、Xamarin)和微软云(Azure)等各种相关技术串联起来的一本书。通过本书读者可以了解到如何利用这些开源、免费的工具和技术构建自己的应用。

本书的代码全部开源,读者有兴趣可访问 <https://github.com/micli/MuscleFellow> 获取本书相关的源代码。同时读者也可以通过访问 <http://musclefellow.chiancloudapp.cn> 来体验已经部署好的网站和 Web API。

本书的创作始于2016年1月,截止到2016年7月2日。期间,.NET Core 从 beta 8 演进到 1.0 RTM。种种变化也给作者写作带来极大的痛苦,很多工作在每次.NET Core 新版本发布之后都要重新调整,很多章节的内容也因此全部失效而永远无法与读者见面。

### 如何高效地阅读本书

对于已经非常熟悉微软战略的读者,可以直接从第3章开始阅读。对于云计算和微软战略不是很熟悉的读者,建议从头开始阅读。

从本书第5章开始,将涉及开发和代码。建议读者从 Github 上下载代码,再与本书内容对照着学习和阅读。

### 本书适合哪些读者

本书适合那些对使用微软技术构建应用以及对微软云平台 Azure 部署应用感兴趣的人员,也适用于对开源技术感兴趣的读者。同时,本书对有志于互联网创业,尤其是期望快速搭建应用的创业者极具参考价值。

### 致谢

首先,我要感谢我的家人,尤其是我的女儿。在女儿需要陪伴成长的年纪,我占用了大量本属于她的时间去写作。尤其是连续写作两本书的情况下,使得我很长一段时间都没有好好地陪她。我对我的女儿充满了愧疚。此外,我也非常感谢我内心深爱的那个人。

其次,我要感谢我的同事刘颖老师和清华大学出版社的盛东亮编辑。本书的成书是和你们的鼓励和支持密不可分的。

最后,我还想感谢胡德民(Peter Hu)和我的老板赵立威。在最近一年的时间里,胡德民帮我安排了很多与微软开源、跨平台相关的技术演讲,使我有机会全方面地接触微软相关的开源技术,尤其是 Xamarin。同时,我的老板赵立威也非常支持我的写作,在写作过程中给予我很大的鼓励和支持。

### 作者联系方式

由于作者能力有限,书中难免会有疏失之处,请读者多多谅解也请不吝赐教。

电子邮件地址: [mic\\_lee2000@foxmail.com](mailto:mic_lee2000@foxmail.com)

微博: @迈克尔-老李

# 目录

## CONTENTS

---

第 1 章 微软“云+端”战略	1
1.1 云计算介绍	1
1.1.1 我们为什么要拥抱云计算	1
1.1.2 当今计算模型的变化	2
1.1.3 云计算是什么	4
1.1.4 什么样的云是一朵好云	5
1.2 微软的公有云 Azure 简介	7
1.3 端的变革	9
1.4 微软的端战略	10
第 2 章 革命性的 ASP.NET Core	14
2.1 完全开源的 ASP.NET Core	14
2.1.1 .NET Core 实现了开源跨平台的使命	15
2.1.2 ASP.NET 是什么	16
2.1.3 ASP.NET 早期开源	19
2.2 ASP.NET Core 的新特性	20
2.2.1 使用 JSON 格式配置文件	21
2.2.2 使用 OWIN 与 IIS 解耦	21
2.2.3 全新的 HttpContext	22
2.2.4 内置依赖注入	23
2.2.5 支持开放的包管理工具	23
2.2.6 采用即时编译运行	24
2.3 MVC 6 的新特性	24
2.3.1 合并技术栈	24
2.3.2 改进路由功能	25
2.3.3 模型绑定	26
2.3.4 TagHelper	27

2.4	dotnet 命令行工具 .....	28
2.4.1	dotnet 命令 .....	28
2.4.2	dotnet-new .....	28
2.4.3	dotnet-restore .....	29
2.4.4	dotnet-run .....	29
2.4.5	dotnet-build .....	29
2.4.6	dotnet-publish .....	30
<b>第3章</b>	<b>开源跨平台的设备端开发 .....</b>	<b>31</b>
3.1	移动开发简史 .....	31
3.1.1	Windows 的精简版 Windows CE .....	31
3.1.2	工控起家的 Symbian .....	32
3.1.3	Android 的革命性崛起 .....	33
3.2	移动开发的显著特性 .....	33
3.2.1	用户体验是第一位的 .....	33
3.2.2	移动设备的资源限制颇多 .....	35
3.2.3	移动应用程序需要处理移动设备的特殊事件 .....	35
3.2.4	移动应用程序需要处理好数据同步和数据冲突的问题 .....	36
3.3	主流跨平台移动开发技术介绍 .....	36
3.3.1	QT Library 介绍 .....	37
3.3.2	Xamarin 介绍 .....	38
3.3.3	Cordova 介绍 .....	39
3.4	移动应用开发方式的选择 .....	40
3.4.1	Web 页面还是 App .....	40
3.4.2	移动应用开发方式的选择 .....	42
3.4.3	Cordova 还是 Xamarin .....	43
<b>第4章</b>	<b>项目介绍和需求分析 .....</b>	<b>45</b>
4.1	项目展望 .....	45
4.2	系统需求 .....	45
4.3	网站功能需求 .....	46
4.3.1	首页功能 .....	46
4.3.2	搜索功能 .....	46
4.3.3	商品详细信息展示 .....	46
4.3.4	购物车功能 .....	47
4.3.5	订单 .....	47

4.3.6	用户账户管理	47
4.4	移动端程序功能需求	47
4.4.1	主界面功能	47
4.4.2	商品信息展示功能	47
4.4.3	购物车功能	48
4.4.4	订单功能	48
第5章	构建开发环境	49
5.1	安装 Visual Studio	49
5.1.1	在云端创建开发机	49
5.1.2	Visual Studio 的版本选择	54
5.1.3	安装 Visual Studio 2015 Community 版本	54
5.1.4	安装小贴士	57
5.2	启用 Visual Studio Online	60
5.3	使用 Azure PowerShell 管理缓存服务	64
5.4	安装 ASP.NET Core 1.0 和 Apache Cordova	65
第6章	构建实体模型	67
6.1	识别实体模型	67
6.1.1	Muscle Fellow 用户关键用例	67
6.1.2	实体类型之间的关系	68
6.2	创建实体模型项目	69
6.2.1	Entity Framework Core 介绍	69
6.2.2	创建 Muscle Fellow.Models 项目	70
6.3	创建实体模型	72
6.3.1	创建用户模型	72
6.3.2	创建商品模型	74
6.3.3	创建商品图片模型	75
6.3.4	创建品牌模型	76
6.3.5	创建商品类型模型	77
6.3.6	创建购物车模型	77
6.3.7	创建订单实体	78
6.3.8	创建订单明细实体	79
6.3.9	创建送货地址实体	80
6.3.10	配置 Muscle Fellow 数据上下文对象	81
6.3.11	将实体对象映射到数据库	82

6.4	创建数据访问项目	84
6.4.1	什么是 Repository 模式	84
6.4.2	创建 Muscle Fellow. Data 项目	85
6.4.3	设计和实现 Repository 模式	86
<b>第 7 章</b>	<b>创建 Web 网站</b>	<b>89</b>
7.1	创建 Web 项目	89
7.1.1	创建 Muscle Fellow. Web 项目	89
7.1.2	ASP. NET Core Web 项目基本结构	89
7.1.3	ASP. NET Core Web 项目的入口	92
7.1.4	引用前端框架	93
7.1.5	引用 NuGet 包	95
7.2	依赖注入	95
7.3	实现商品展示相关功能	97
7.4	实现首页功能	102
7.5	实现购物车的相关功能	104
7.6	实现收货地址管理相关功能	108
7.7	实现订单相关功能	113
<b>第 8 章</b>	<b>创建 Web API</b>	<b>119</b>
8.1	几种远程调用方式	119
8.1.1	套接字调用	119
8.1.2	RPC 服务调用	120
8.1.3	Web Service 调用	121
8.1.4	Web API 调用	121
8.2	设计好用的 Web API	122
8.2.1	Web API 应具有版本标识	122
8.2.2	Web API 应简单明了	122
8.2.3	选择恰当的数据格式	123
8.3	Web API 的配置和验证	124
8.3.1	实现 ASP. NET Core 配置	124
8.3.2	Web API 的用户身份验证	125
8.4	创建 Muscle Fellow. API 项目	128
8.5	设计 Muscle Fellow 的 Web API	131
8.5.1	商品实体 Web API 的设计	131
8.5.2	商品实体 Web API 的实现	132

8.5.3	商品实体 Web API 的调试	136
8.5.4	用户账户实体 Web API 的设计	139
8.5.5	用户账户实体 Web API 的实现	140
8.5.6	购物车实体 Web API 的设计	143
8.5.7	购物车实体 Web API 的实现	144
8.5.8	订单实体 Web API 的设计	146
8.5.9	订单实体 Web API 的实现	146
<b>第 9 章</b>	<b>在 Azure 上部署服务</b>	<b>149</b>
9.1	在 Azure 上部署示意	149
9.2	在 Azure 上部署的注意事项	150
9.2.1	永远要为虚拟机附加一块磁盘	150
9.2.2	故障域和可用性集	151
9.2.3	SQL Azure PaaS 服务	151
9.3	在 Azure 上创建虚拟网络	152
9.4	在 Azure 上创建 Linux 虚拟机	154
9.5	为 Linux 虚拟机附加磁盘	159
9.6	在 Ubuntu 上部署 .NET Core	162
9.7	迁移 SQL 数据库到云端	163
9.8	部署 Muscle Fellow. Web 和 Muscle Fellow. API	169
9.8.1	发布前修改配置	169
9.8.2	制作和部署发布包	169
9.9	使用 Nginx 实现反向代理	170
9.10	设置虚拟机端口映射	173
<b>第 10 章</b>	<b>AngularJS 和 Ionic Framework</b>	<b>176</b>
10.1	AngularJS 入门	176
10.1.1	Angular 基本概念	176
10.1.2	AngularJS 模块	177
10.1.3	AngularJS 的作用域	178
10.1.4	控制器	178
10.1.5	依赖注入	178
10.1.6	HTML 页面中加载 AngularJS	179
10.1.7	AngularJS 的全局 API	179
10.1.8	AngularJS 的数据绑定	180
10.2	Ionic Framework 入门	181

10.2.1	模态视图 .....	181
10.2.2	SideMenu 侧面菜单 .....	183
10.2.3	导航 .....	183
10.2.4	列表控件 .....	184
<b>第 11 章</b>	<b>使用 Cordova 开发跨平台移动应用 .....</b>	<b>186</b>
11.1	创建 Ionic 项目 .....	186
11.2	开发 Cordova 版本移动应用 .....	187
11.2.1	Ionic 项目结构介绍 .....	187
11.2.2	Ionic 项目的启动过程 .....	188
11.2.3	实现商品列表界面 .....	191
11.2.4	实现商品详细信息界面 .....	193
11.2.5	实现用户登录 .....	197
11.2.6	实现购物车功能 .....	198
11.3	项目编译调试 .....	200
11.3.1	Cordova 项目适用的调试器 .....	200
11.3.2	项目调试策略 .....	203
11.3.3	测试小贴士 .....	204
11.3.4	在 OS X 上编译 .....	204
<b>第 12 章</b>	<b>使用 Xamarin 开发跨平台移动应用 .....</b>	<b>207</b>
12.1	准备工作 .....	207
12.1.1	安装 Xamarin .....	207
12.1.2	配置苹果 OS X 系统 .....	209
12.1.3	实现 Visual Studio 和 OS X 系统的连接 .....	210
12.2	开发 Xamarin 版本移动应用 .....	211
12.2.1	创建 Xamarin 应用程序项目 .....	211
12.2.2	Xamarin 项目结构介绍 .....	212
12.2.3	认识可移植项目 .....	215
12.2.4	实现商品列表界面 .....	217
12.2.5	iOS 项目构建和调试 .....	220

## 第 1 章



# 微软“云十端”战略

本章将带领读者概览一下微软这些年在云和端上做的工作,以及“云十端”的模式究竟会给 IT 行业带来什么样的影响。

## 1.1 云计算介绍

云计算,是最近一个非常热的词汇。作为一个 IT 人士要是不跟人家聊聊云计算,都不好意思跟别人打招呼。那么云计算到底是什么呢?本节将带你了解什么是云计算。

### 1.1.1 我们为什么要拥抱云计算

当今的世界发展日新月异。对于 IT 应用来说,十几年前提出的“企业级”应用和“电信级”应用已经不再适应当今业务发展的需要了。我们先来了解一下什么是“企业级”应用计算。所谓企业级的计算,一般来说是指能够在企业工作时间内,保障稳定高效运行的计算系统。典型地,比如,企业内的人力资源系统、供应链系统、财务系统等。这些系统的用户都是企业内部员工和企业的供应商。因此,我们在构建这些系统时可以明确地知道应用的用户数量和用户使用这个系统的主要时间段。例如,对于企业的财务系统来说,用户就是企业的财会人员,数据量根据企业每月交易笔数也非常好判断,这样的系统只要保证在企业正常工作时间(最多早 8 点到晚 8 点)正常运行就够了。遇到业务忙时,比如月初和月末的几天,适当增加保障和运维的力度也就可以了。因此,当时很多软件标榜的“企业级”在现在看来也不是什么高标准的系统。

随着全球经济一体化,和 IT 对社会变革的深刻影响。以前的“企业级”计算将不再适应当前的业务需求。这主要表现在以下几个方面。

对于全球化的企业,企业的 IT 系统也需要全年无休且稳定地运行了。现在很多企业都进行了跨国并购和在海外设立分支机构。企业员工的工作时间覆盖了地球的全部时区,往往是亚洲的员工下班,欧洲的员工开始工作,欧洲的员工下班,美洲的员工开始工作这样周而复始。每个工作周留给业务系统运维的时间窗口极其有限。

随着企业面临更多的业务挑战,IT 系统必须随之迅速变化为企业的业务提供有力支



持。比如我们熟知的 SOHO 中国有限公司,它从以前一家专营房屋建造的房地产公司转换为持有优质房产资源的房产租赁公司。这由卖转租的转变,将会给其 IT 系统带来极大的挑战。它的 IT 系统现在要记录公司持有的各项物业的精确信息,如租房合同、租赁到期信息等内容。这比以前售卖房产要复杂得多。但是,IT 系统不能拖公司转型的后腿。因此,需要快速地实施 IT 系统的转变。

随着人力成本增加,企业的 IT 成本也在与日俱增。以前企业构建一套系统就安排专人维护的时代将一去不返。企业迫切需要易于部署、低成本、高效运行的 IT 系统。

这一切的挑战,都需要一种全新的计算方式来革命性地解决。云计算就是一种革命性的计算方式。

### 1.1.2 当今计算模型的变化

随着企业需求的转变,计算模型也随之产生了深刻的变化。以下是目前几种典型的计算模型。

模型一:开关式计算模型(见图 1.1)。

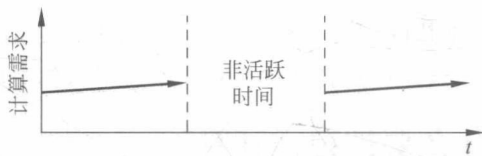


图 1.1 开关式计算模型

这种计算模型的特点是服务只在特定的时间段开启,应用需要在服务开启的时间段内进行高等级的维护保障。举例来说,每年公务员招考报名、研究生招考报名和各种选秀投票都属于这种类型的计算应用。在计算服务开启时,要严格保障数据的准确性、应用的稳定性以及大开发情况下的执行效率。另一方面这些应用并不强调每天一直不停顿地运行。

从宏观的方面来说,这种类型的应用只在一定的时间段内对计算资源有需求。从财务的角度来说,这是相当不划算的。因为服务器和其他硬件一旦购置了就开始计算折旧,无论使用与否都会计入成本。在计算资源闲置的时候,在资金财务成本上与一直保持运行的应用差不多。

模型二:快速增长模型(见图 1.2)。

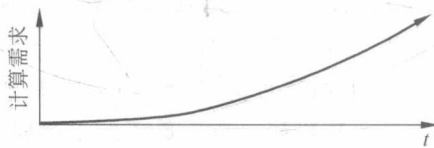


图 1.2 快速增长模型