HZ BOOKS
华章科技
PEARSON

云计算技术系列丛书

构建云应用

概念、模式和实践

Building Applications in the Cloud

Concepts, Patterns, and Projects

（美）Christopher M. Moyer 著
顾毅 等译

机械工业出版社
China Machine Press

构建云应用

概念、模式和实践

Building Applications in the Cloud

Concepts, Patterns, and Projects

（美）Christopher M. Moyer 著
顾毅 等译

机械工业出版社
China Machine Press

本书沿用大家所熟悉的“设计模式”格式，介绍了云计算平台上行之有效的模式。通过基于Python和Amazon Web Services（AWS）平台的详细的范例代码和应用程序，向读者演示了这些模式的实际运用。本书内容包括掌握成功的云计算解决方案的核心原则，如何正确地构建软件即服务（Software as a Service，SaaS）模式，理解不同的云服务提供商所提供的服务，如何设计主机镜像、为云平台上的应用打造一个坚实的基础，如何针对与外部系统的交互行为选择最合适的模式，如何实现高效的数据处理以及如何充分发挥多主机集群部署的优势。

本书适合软件开发人员和对云计算感兴趣的读者阅读。

本书版权登记号：图字：01-2011-3368

图书在版编目（CIP）数据

构建云应用：概念、模式和实践 /（美）莫耶（Moyer, C. M.）著；顾毅等译. —北京：机械工业出版社，2012.3

（云计算技术系列丛书）

书名原文：Building Applications in the Cloud: Concepts, Patterns and Projects

ISBN 978-7-111-37312-4

Ⅰ. 构…　Ⅱ. ① 莫…　② 顾…　Ⅲ. 计算机网络　Ⅳ. TP393

中国版本图书馆CIP数据核字（2012）第015510号

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：关　敏

北京市荣盛彩色印刷有限公司印刷

2012年3月第1版第1次印刷

186mm × 240mm · 16.5印张

标准书号：ISBN 978-7-111-37312-4

定价：49.00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）88378991；88361066

购书热线：（010）68326294；88379649；68995259

投稿热线：（010）88379604

读者信箱：hzjsj@hzbook.com

译 者 序

今天，个人电脑、移动设备、智能手机正以前所未有的速度爆发式地增长着。这些小型化的电子设备之所以能快速发展，完全得益于因特网，尤其是Web 2.0的发展。由于这些设备的计算能力有限，所以计算需求都渐渐转移到了服务器上，人们通过网络使用远程的计算结果，并通过这些轻量化的设备获取远程服务器上的服务和资源。这也是今后人们使用软件的一大趋势。

那么作为服务的提供者，该如何才能拥有足够的运算能力来为用户提供服务呢？过去，人们会使用性能强大的服务器来实现这个需求。然而，随着硬件制造越来越接近工艺上的极限，30多年来，业界一直遵循的摩尔定律也许在未来的几年内就会成为历史。然而，近几年一种称为“云计算”的概念诞生了，与传统的使用计算资源的方式不同，云计算将计算资源视为一种服务，形同公用事业一般的服务。向最终用户提供服务的公司再也不用事先通过估算去采购昂贵的硬件了，取而代之的是他们可以按需使用这些取之不尽用之不竭的计算资源。

于是，一个全新的时代到来了！

Amazon、Google等行业巨头都先后推出了自己的云服务，与此同时，许多小型的云服务提供商也如雨后春笋般涌现出来。根据服务的不同类型，云服务可分为IaaS（基础设施即服务）以及PaaS（平台即服务）两大类。使用云平台和服务需要大量的经验和技巧。我在工作中也使用过IaaS型的Amazon Web Services。当时，由于缺乏经验，也只是将EC2当做单纯的虚拟机来使用。如此使用云平台的公司或个人其实也不占少数。但是，随着使用的深入，我们发现，如果仅仅是这样来使用云平台，完全无法发挥出云平台的自动伸缩、按需分配资源等优势。而且，云的计费系统普遍比较复杂，如果不注意使用技巧，往往会产生高额的账

单，这完全掩盖了云平台的好处。我们不禁对这个全新的时代产生了不少疑惑。

幸运的是，作者在进行了大量开发，积累了大量经验之后，为我们带来了这本书。本书介绍了许多云计算的核心概念，以及不同的云服务提供商所提供的云平台的特色。更为重要的是，本书用很大的篇幅讲解了在云平台上开发应用程序的模式和技巧。就像GOF的设计模式那样，书中所介绍的这些模式对我们解决在日常工作中所遇到的问题具有非常实际的指导意义。此外，本书最后还展示了两个项目的范本，并将书中所介绍的概念和技巧运用到其中，给读者十分直观且深刻的印象。本书做到了理论和实践的平衡。对于每一位云平台的开发者来说，本书具有非常高的实用价值。

希望各位读者看完本书之后，对云计算能有更进一步的理解，能够开发出更好的应用。

由于云计算的发展快速很快，书中提到的一些服务和工具在翻译过程中已经发生了一些变化，我在书中用“译者注”的方式进行了说明。

参与本书翻译和校对工作的还有曾燕萍。在此，我也要感谢机械工业出版社的陈冀康老师和关敏老师以及其他参与本书编辑和出版的工作人员，感谢大家给予我的帮助和为本书所付出的辛勤劳动！翻译过程中难免存在疏漏，也请各位读者多多包涵并予以指正。

前　言

作为一家小型创业公司的开发人员，做了几个月将既有服务移植到云环境的开发工作之后，我开始意识到这些工作由我一个人来做的话工作量实在太大了。于是，我开始寻找其他的开发人员来协助我，或者当我找到更好、更有意思的工作机会时让他顶替我。不久，我就发现真正彻底理解开发基于云平台的应用程序所必需的复杂度的人相当少，而且这些人几乎都对自己目前的公司很满意。

就如何使用云平台的话题，我开始在博客中写专题文章，网址是http://blog.coredumped.org，但我很快又意识到，我可以花上整整一年写下所有这些人们应该知道的东西。这些文档应该更适合放在参考书中而不是随随便便地散落在博客各处，所以我决定写一本书。

本书的目的

本书并不是一本需要从头至尾去读完的教程，而是一本教你如何在云环境中创建应用程序的指南，也是一本可以在遇到特定问题时随时参考的手册。当公司为你安排新的项目，并告诉你要让它具有伸缩性时，看看本书中所讨论的模式是不是适用于该项目。当你在项目中发现了特定的问题却束手无策时，翻一翻本书。如果你尝试开始一个新的项目，并且你有一个非常完美的想法，却不知道该如何让系统能够收缩自如时，翻一翻本书。如果你正尝试改造一个既有的项目，使其能在云环境中自由扩展时，翻一翻本书。如果你不知道在云计算环境中能构建什么样的应用程序时，翻一翻本书。

本书中并没有发明许多新的模式，而仅仅是向你展现了一些技巧和新技术，这些都是在

云环境中运行系统时所需要考虑的。尽管你可以将书中讨论的模式用于任何形式的集群环境，但这些模式是为云计算所提供的服务量身定做的。

如何使用本书

本书分为三个部分。每个人都应该阅读第一部分，从而对云计算有一个基本的理解。在第二部分中，你可以挑选最感兴趣的模式阅读。如果你从未开发过任何形式的云应用程序，建议看一下第三部分中的范例应用程序，从而使你对该系统究竟最适合于什么样的应用程序有一个确切的了解。

第一部分："概念"

第一部分旨在让你掌握一些在云平台上进行开发的基本概念。你应该全部阅读这一部分，该部分分为以下几章。

第1章："云服务的基础"——提供了一些使用云计算解决方案时的基本概念，对于任何对云计算平台感兴趣的开发人员来说，本章是绝不容错过的。

第2章："把软件做成服务"——针对如何实现软件即服务（SaaS）提供了一些基本的指南。本章详细讨论了为什么软件即服务是个好主意，以及如何正确构建SaaS。

第3章："云服务提供商"——提供了一些由不同云服务提供商所提供的服务的实例。

第二部分："模式"

第二部分更像是一本手册，给出了构建系统时的问题以及相应的解决方案。

第4章："设计镜像"——包括基本的构建基础镜像的模式，这些镜像是构建应用程序的基础。

第5章："设计架构"——包括让系统与外部系统（不是你所采用的云服务提供商的系统）进行交互的模式。

第6章："操作数据"——包括针对数据执行代码片段的模式。

第7章："集群系统"——包括为了充分发挥集群部署优势而设计的基础框架中使用的模式。

第三部分："项目"

第三部分以真实世界中的应用程序为例，并将书中所介绍的模式运用进去。这两章都采用了相同的入门教程，但实现的方式不同。

第8章："简单的博客系统"——详细介绍了如何从头开始构建一个简单的博客系统，且不采用任何既有的框架。

第9章："使用Marajo开发博客系统"——详细介绍了如何使用Marajo这一基于云环境的Web应用框架来构建博客系统。

从何开始

现在，许多人的第一个问题就是：从何开始？该如何迅速开始开发应用程序？如果不想完整地阅读本书，怎样做才能简单快速地了解到书中这些概念的实际工作原理？

选择了本书，你就已经走上了正确的轨道。如果随随便便地选择一家云服务提供商，启动几台服务器，期待能从这些服务商那里得到你所想要的东西，毋庸置疑，这是不切实际的。如果只是挑选一家云服务提供商，而不做足够的事前研究，其结果往往会让人们陷入一大堆问题中，而且还会把这些问题归咎于云服务提供商。这就相当于你买了一辆手动挡的汽车，但完全不了解该如何驾驶，结果却向销售商抱怨说是他们把车卖给了你。如果你在事前没有做研究或准备，那么在云环境中遇到问题时，完全不应该感到意外。如果你不是开发人员，找第三方来帮你管理云环境可能会更合适，但如果你阅读了本书，我想你绝对不会满足于"找一群家伙来搞定它"。

致谢

我要感谢我的伙伴和导师Mitch Garnaat，感谢他给予我的帮助和激励，让我进入了云计算领域。我也要感谢Amazon Web Services的团队，感谢他们推动云计算的市场不断向前，且持续不断地带来新的产品，使本书中的内容成为可能。

关于作者

Chris Moyer毕业于美国罗切斯特理工大学（Rochester Institute of Technology，RIT），并获得软件工程学士学位。Chris拥有5年以上的编程经验，主要集中在云计算领域。他的大部分时间都在开发大受欢迎的boto客户端类库，该类库用于与Amazon Web Services进行通信。通过向boto的创始人Mitch Garnaat学习，Chris随后开始基于该客户端类库开发Web框架，也就是Marajo和botoweb。基于这些框架，他还创建了大规模的应用程序。

Chris目前是Newstex公司的技术部副总裁，负责管理技术研发工作，这些技术用于将应用程序迁移到云平台，同时他也管理着自己的部门，该部门正在积极地维护和开发几个应用程序。Chris与妻子Lynn住在纽约。

目　录

第二部分　模式

第0章 引 言

在深入开发云应用之前，你需要了解云计算背后的几个关键性概念。“云计算”(cloud computing) 这一术语出现在我们身边不过才短短几年的时间，但是，关于如何使用它的概念和模式从云计算时代的黎明时期就已经为人所用了。

0.1 云计算是什么

关于云计算有上百种字面上的定义，其中大多数对于这些定义的创造者以外的普通人而言，很难甚至完全无法理解。许多公司把他们的虚拟主机环境称作云 (cloud)，因为这意味着强大、高速、可扩展性。实际上，云相当于那些拥有无限扩展能力的计算和存储资源的集群系统。大多数云都允许按实际使用情况计费。

把应用部署在云计算的环境中运行，其典型的优势就是成本低廉，因为对于大多数服务而言，你无须前期投资，也无须为昂贵的IT人员支付费用。虽然这是全面切换到云计算服务的绝佳理由，但并不仅限于此。对于运用合理的设计模式构建的应用，云计算能赋予你对其进行实时扩展的优势。此外，对于多年来精心管理系统的人员，云计算还能减少他们的工作负荷。一支训练有素的团队在应对需求增长方面需要花费更多的时间，因为他们必须外出采购硬件。反过来，当采用云平台时，你只需简单地从巨大的可用资源池中申请更多的硬件使用时间。由于这些云服务通常是由更大的公司提供的，因而他们有实力雇佣一组人员不分昼夜地确保系统以最高性能运行。

如果你曾有过在深夜不得不为失效的服务器恢复运行的经历，那么你一定知道找人去恢复系统是多么痛苦的一件事。把物理服务器搬迁到云中从而降低负载，你就能直接专注于你的应用，而无须再为系统管理担心了。

企业家的目标是与数百万的顾客发生不断激增的业务，但是大多数人都没有足够的原始资本来构建一个能针对潜在增长而灵活扩展的服务器场 (server farm)。如果你们只

是个创业型企业，那么你甚至都不知道能不能成功。但是假设你的公司突然间成功了，现在你必须迅速扩展你的Web应用，在一夜间从仅处理20个用户增长到能够处理20 000个用户。

在需要自己准备服务器的传统环境里，这可能意味着你不仅需要为服务器场添置更高效的管线，而且还需要购买全新的更快的服务器，并把它们组建起来，期望它们能在一起顺利工作。然后你就要开始担心数据管理以及与管理集群系统相关的各种有趣的事情。如果你用过云，那么你只需要简单地向你的云服务商发出几次API调用，几分钟内你的应用就能获得40台新的机器资源。一周后，倘若用户数量减少到了200，你可以关闭一些闲置的服务器实例。这能帮你节省开支，因为你不必真正买下它们。

0.2 云计算的革命

稍后会提到，云计算和其他新的软件架构的设计与过去的设计有多少相似之处，也许我们的确是倒退了。尽管与过去完成的设计相比有些相似之处，但设计技巧却着实有着很大的不同。下面，我们对应用开发的历史做个简短的回顾。

0.2.1 主机

最早出现的是主机（mainframe）。主机是单个的超级电脑，拥有当时市面上最快的处理芯片。它们是昂贵的商品。主机是个庞然大物，需要大量的冷却和电力，以至于甚至连大公司通常都不会自己购置一台主机去用，所以，你不得不通过哑终端（dumb terminal，一种特殊的电脑，除了与主机连接外，几乎没有别的功能）从远程与它们通信，如图0.1所示。后来这些主机变得越来越小，但是你仍必须使用瘦客户端与它们交互。

这些主机设计用于快速地执行单个任务，所以那时根本无须考虑什么并行处理，一切都是简单地按顺序完成。主机需要一个复杂的调度系统，负责给每项处理分配一定的时间，并允许其他处理任务在主机的工作间隙插入请求。

与最原始的只有一个接口与计算机相连的系统相比，这种客户端-服务器交互方式是一个巨大的进步。在很多场合，这仍然是广泛采用的架构，虽然哑终端已经被瘦客户端所取代。瘦客户端是设计用于在普通电脑上运行的软件，除了简单地与主机连接外，

并没有很多其他功能。Web浏览器就是至今仍在使用的瘦客户端的一个绝佳例子。

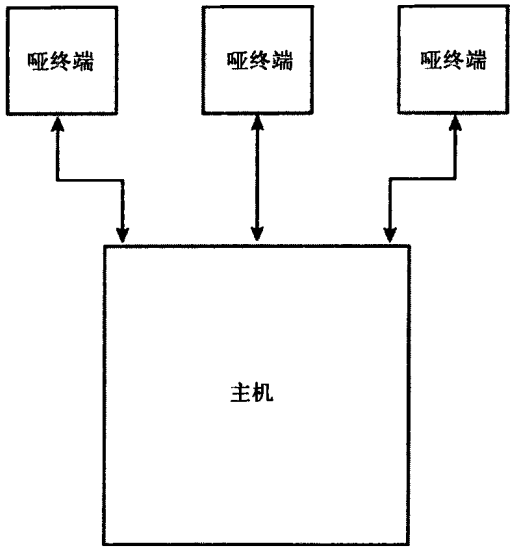

图0.1 主机

0.2.2 PC革命

虽然最早的主机很健壮，但实际上它们的能力却比不上当今的一块数字腕表。后来，技术革新使高性能的主机成为历史，并进化出了更小更强大且真正能装进普通大小的屋子里的设备。这些新设备对软件构建方式发起了一场革命，使应用开发人员能在客户端系统中实现任何事情。这意味着你不再受制于网络连接速度的瓶颈，无须把网速问题考虑在内，你所见到的速度迟缓都源自于你的个人电脑。机器越来越快，软件则不断对本地机器提出更高的要求。今天的PC机，对于大多数日常任务来说已经绰绰有余了。这使得多任务产生了有趣的前景，单一的系统可以同时执行多个任务。以前，任务的执行是简单地由主机中使用的更为先进的调度系统处理的，但后来，被硬件线程甚至多处理器系统所取代。

许多开发者拒绝适应全新的多任务技术，但有一些则接受了。后者开发出了可以在单一处理系统的时间碎片中运行的大规模系统，充分利用了硬件资源。这对下一个技术上的跨越——高速互联网起到了帮助作用。