Vert.x

应用开发实例教程

吕海东 张坤 编著

清华大学出版社

Vert.x
应用开发实例教程

吕海东 张坤 编著

清华大学出版社
北京

内 容 简 介

Vert. x是一个轻量级的高性能JVM应用平台，基于它可开发各种移动、Web和企业应用服务器端应用。

Vert. x主要特点是可使用多种语言编写应用，如Java、JavaScript、CoffeeScript、Ruby、Python或Groovy等。

Vert. x的简单actor-like机制能帮助脱离直接基于多线程编程，该机制是基于Netty和Java 7的NIO2的编写的。

Vert. x的工作模式与Node. js基本相同，其目的在于为JVM提供一个Node. js的替代方案，二者都采用非阻塞的异步工作模式。所有的Vert. x组件Verticle都工作在一个Event Loop单线程内。并且所有Verticle部件之间完全非耦合，它们之间不能直接调用，只能通过Event Bus发送和接收事件Event完成相互的调用和数据通信。

本书全面采用案例驱动，主要知识的讲解都辅助以实际案例应用编程，便于读者的理解和自主学习和运用。知识讲解通俗易懂，详略得当，重点突出。

本书每章都附以PowerPoint课件来总结本章中的大纲和重点内容，便于教师教学和读者复习和理解。

本书旨在为Vert. x的初学者和大中专院校学生提供易于入门，全面了解和掌握Vert. x框架技术和应用的教材和辅导资料，为使用Vert. x开发实时应用和企业级应用打下良好的基础。

图书在版编目(CIP)数据

Vert. x应用开发实例教程/吕海东，张坤编著. --北京：清华大学出版社，2015
ISBN 978-7-302-41629-6

Ⅰ. ①V…　Ⅱ. ①吕… ②张…　Ⅲ. ①网站－软件开发－教材　Ⅳ. ①TP393. 07

中国版本图书馆CIP数据核字(2015)第228394号

责任编辑：付弘宇　柴文强
封面设计：何凤霞
责任校对：李建庄
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
质 量 反 馈：010-62772015，zhiliang@tup. tsinghua. edu. cn
课 件 下 载：http://www. tup. com. cn，010-62795954

印 装 者：北京国马印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：20. 5　　**字　　数**：509千字
版　　次：2015年12月第1版　　**印　　次**：2015年12月第1次印刷
印　　数：1～2000
定　　价：45. 00元

产品编号：064703-01

目录

CONTENTS

第1章

Vert.x概述

本章要点

- Vert. x 的概念。
- Vert. x 诞生的背景。
- Vert. x 的特性。
- Vert. x 的安装与测试。
- 简单 Vert. x 应用的编程与运行。

本章重点介绍了 Vert. x 的概念、诞生的背景、主要特性以及 Vert. x 平台的安装和启动。通过本章的学习读者基本能对 Vert. x 有基本的了解，理解 Vert. x 的应用领域，以及 Vert. x 与其他平台的比较和区别。

1.1 Vert. x 的概念

按照 Vert. x 的官方定义，Vert. x 是基于 JVM(Java Virtual Machine)的、轻量级、高性能的用于开发现代移动应用、Web 应用和企业级应用的服务器平台。

1.2 Vert. x 诞生的背景

当前由于移动互联网的飞速发展，智能手机和平板请求访问 Web 应用极为普遍，进而产生大量移动客户端访问 Web 应用导致 Web Server 无法满足如此巨大的高并发性访问需求，即所谓的 C10K 问题，当并发连接超过 10 000 以上时使用传统技术会引发暂停，类似 QQ、微信的实时数据传输类应用表现尤为突出。

传统的企业级 Web 应用一般使用基于 Oracle 公司的 Java EE 的 JSSHA(Java EE＋Struts＋Spring＋Hibernate＋AJAX)框架[3]，或基于微软 DOTNET 平台开发，它们都使用多线程技术实现多连接请求，因此都面临 C10K 性能瓶颈，要解决大并发客户的请求只能进行服务器群集，导致系统设备投资巨大，中小企业难以承受。

在 Java™和 PHP 这类语言中，每个连接都会生成一个新线程，每个新线程可能需要 2MB 的配套内存。在一个拥有 8GB RAM 的系统上，理论上最大的并发连接数量是 4000

个用户。随着您的客户群的增长，如果希望 Web 应用程序支持更多用户，那么就必须添加更多服务器。当然，这会增加服务器、流量和人工等成本。除这些成本上升外，还有一个潜在技术问题，即用户可能针对每个请求使用不同的服务器，因此，任何共享资源都必须在所有服务器之间共享。鉴于上述所有原因，整个 Web 应用程序架构(包括流量、处理器速度和内存速度)中的瓶颈是：服务器能够处理的并发连接的最大数量。

为解决以上问题，最先出现的解决方案就是 Node.js 技术，通过采用单线程、非阻塞、异步工作模式、事件循环机制很好地解决了 C10K 问题，一般情况下单 CPU 服务器即可支持超万个并发连接。Node 解决这个问题的方法是：更改连接到服务器的方式。每个连接发射一个在 Node 引擎的进程中运行的事件，而不是为每个连接生成一个新的操作系统线程(并为其分配一些配套内存)。Node.js 声称它绝不会死锁，因为它根本不允许使用锁，它不会直接阻塞 I/O 调用。Node.js 还宣称，单服务器能支持数万个并发连接。

Vert.x 在 Node.js 的优点之上，将基础平台移到 JVM 上，引入分布式并发机制以及自动支持多核 CPU 的能力，在一个单 CPU 八核的 Intel E5 服务器上即可支持数十万并发连接请求。图 1-1 展示了单线程情况下，不同语言编写的 Vert.x 与 Node.js 在处理 HTTP 请求生成状态码 200 的性能比较，从图 1-1 可见 Vert.x 每秒能处理的 HTTP 请求个数要远远超越 Node.js，即使使用最慢的 Ruby 语言编程的 Vert.x 也超过每秒 20 万个请求，Java 实现的 Vert.x 每秒处理几乎接近 33 万个请求，即使是六个线程的 Node.js 也无法与单线程的 Vert.x 相比，可见 Vert.x 的性能之卓绝。

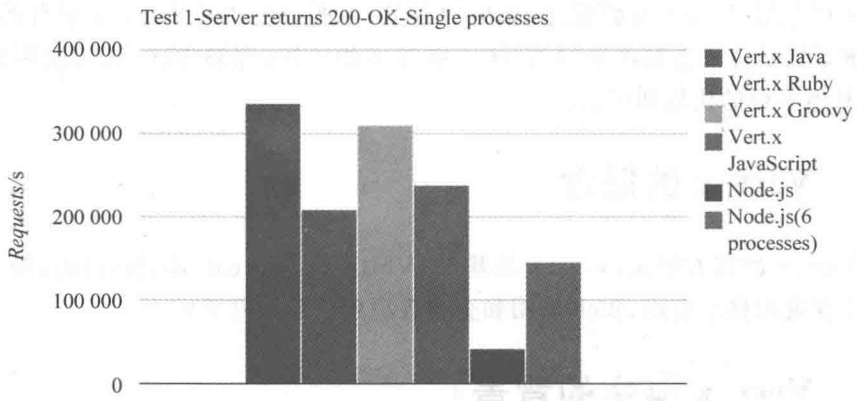

图 1-1　单线程下 Vert.x 与 Node.js 的性能比较

图 1-2 展示了同样实现 Web 服务器处理静态页面请求时，单线程下 Vert.x 与 Node.js 的性能比较，几乎所有语言实现的 Vert.x 每秒都能处理超过 9 万个请求，而 Node.js 则没有超过 3 万的，即使六线程下的 Node.js 也无法与单线程的 Vert.x 相比。

Vert.x 全面超越 Node.js 主要体现在以下几点：

(1) 性能：对于很多应用程序，Vert.x 可以在性能上轻松击败 Node.js。与 Vert.x 1.x 相比，Vert.x 2.x 有许多性能改进(在许多基准测试中，它都比 Node.js 更快)，而且还充分利用了 JVM——一个无与伦比的虚拟机——的能力。Vert.x 与 Netty 团队密切合作，作为 Vert.x 核心开发人员之一的 Norman Maurer 也是 Netty 的核心开发人员，使得 Netty 4.0 用在了 Vert.x 2.x 版本中，为 Vert.x 带来了巨大的性能提升。

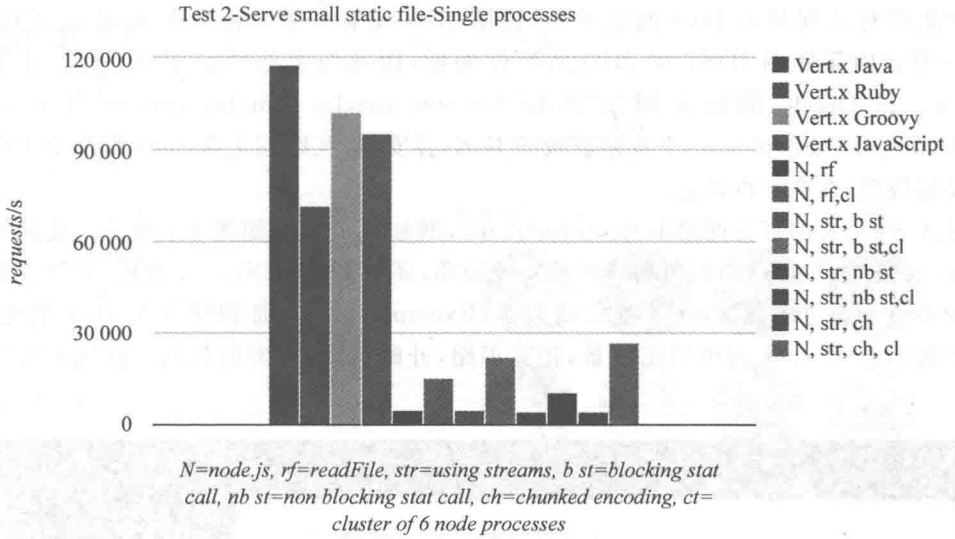

图 1-2 单线程下 Vert. x 与 Node. js 实现 Web 应用静态文件请求性能比较

(2) 语言选择：Node. js 只支持 JavaScript，JavaScript 并不是对任何应用开发都适合，每种编程语言都有自己的强项和最佳应用领域。Vert. x 支持 Java、JavaScript、Ruby、Groovy 和 Python，新版本将支持 Scala 和 Clojure 等并发编程语言。

(3) 功能扩展：实际上 Vert. x 远不止是“JVM 的 Node. js”，Vert. x 要大很多，它是一个多语言的、分布式应用程序平台。在某种程度上，可以认为 Node. js 所做的是 Vert. x 的一个子集，但 Vert. x 的目标更广。

1.3 Vert. x 的安装

Vert. x 平台运行在 Java 虚拟机(JVM)上，而 JVM 是跨平台的，因此 Vert. x 也支持各种服务器平台，在 Windows、OS、Linux、UNIX 等上都可以运行。

安装 Vert. x 之前，首先检查是否安装了 Java 开发环境 JDK，通过 cmd 指令启动命令行窗口，再输入 java -version，检查 JDK 是否安装，参见如图 1-3 所示的检查 JDK 安装操作窗口。Vert. x 2. x 需要 JDK7，而未来 3. x 版需要 JDK8 才能运行。

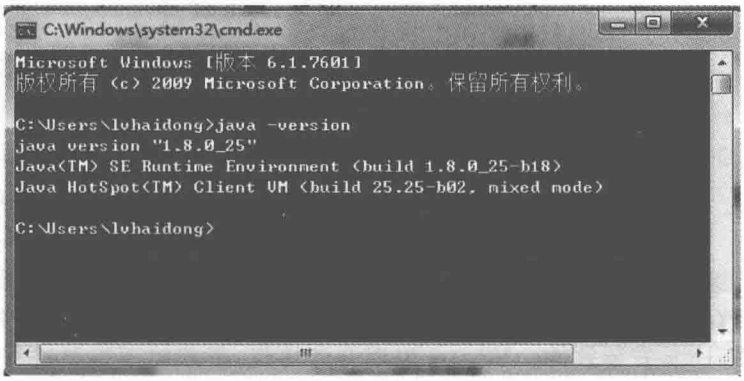

图 1-3 检查 JDK 是否安装截图

如果没有出现显示 Java 的版本号，说明 JDK 没有安装。需要首先安装 JDK，并且 Vert. x 平台内部使用 JDK7 的 NIO 新特性编程，因此要求必须是 JDK7 以上才能运行 Vert. x。到 Oracle 的官方网站（http://www. oracle. com/technetwork/java/javase/downloads/jdk8-downloads）下载最新版的 JDK，并安装，互联网上和 Java 编程书中都有详细的安装说明，在此不再赘述。

进入 Vert. x 的官方网站 http://vertx. io/，网站主页显示如图 1-4 所示。此网站是学习 Vert. x 最佳的地方，不但包括详尽的安装说明，还有齐全的 Vert. x 文档，各种支持语言的编程手册。通过下载 Vert. x 提供的案例（Examples），可以加快学习 Vert. x 的进程，使我们尽快熟悉 Vert. x 的可应用场景，拓宽思路，开阔视野，开发出我们自己创新性的应用项目。

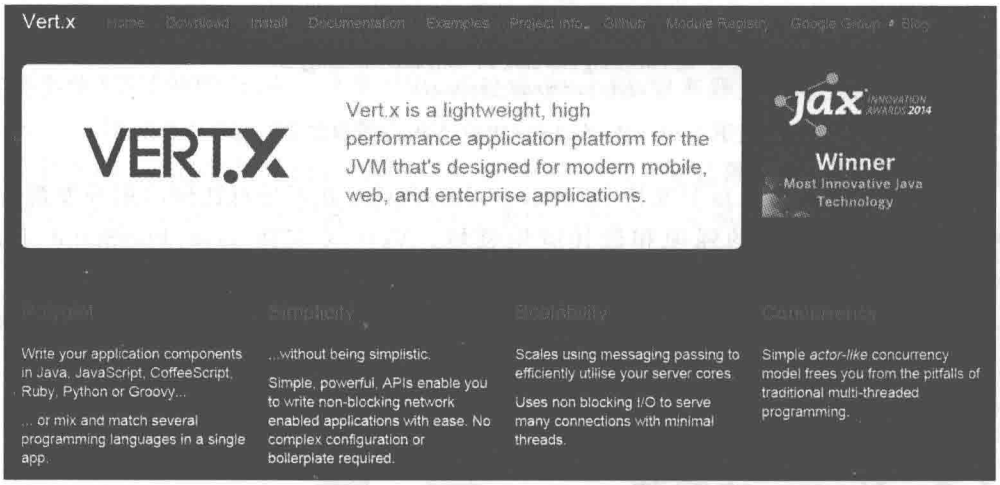

图 1-4　Vert. x 官方主页

在顶部主导航中点击 Download 下载链接，进入 Vert. x 版本选择页面，编写本书时，最新的版本是 2. 1. 5，如图 1-5 所示。

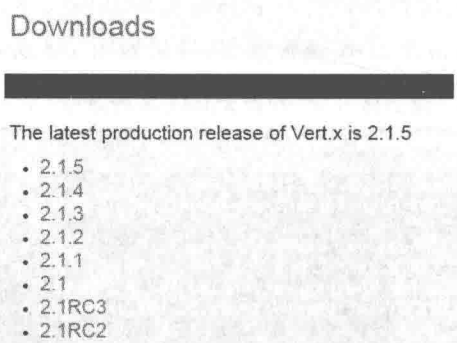

图 1-5　Vert. x 版本选择

选择最新的 2. 1. 5，进入如图 1-6 所示的下载页面，单击 Downloads 下面的 vert. x-2. 1. 5. zip 链接即可下载 Vert. x 安装软件。Vert. x 采用开源协议，下载时不需要进行用户的注册或登录。

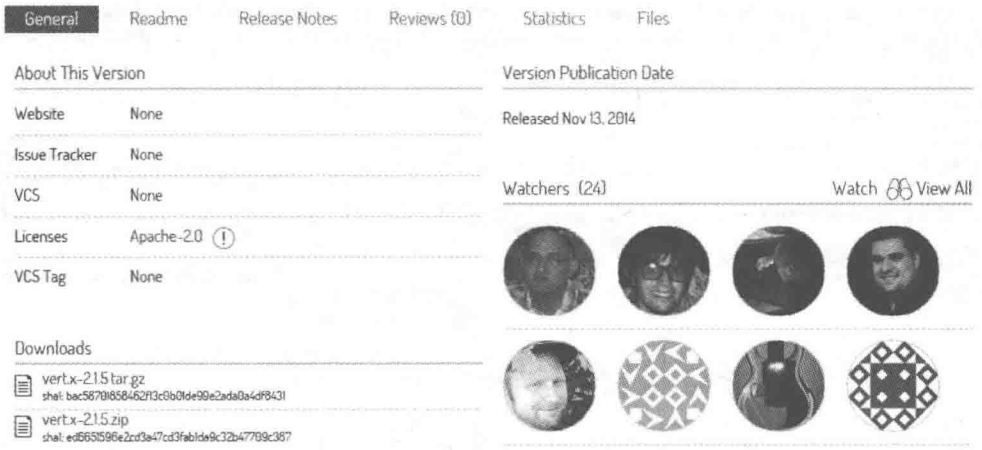

图 1-6 Vert.x 下载页面

下载后的 zip 压缩文件，直接解压到任意指定的目录，目录最好不要包含汉字和空格，本书把 Vert.x 解压到 D:\apps\vertx215 目录中，如图 1-7 所示。

图 1-7 Vert.x 的安装目录

Vert.x 的运行需要将 bin 子目录配置到操作系统的 path 环境变量中，以便能启动 Vert.x 平台。

在 Windows 中，通过开始→控制面板→系统→高级系统设置→环境变量→path→编辑，将 Vert.x 安装目录下的 bin 子目录配置到 path 变量中，如图 1-8 所示。

在配置好 Vert.x 的 path 变量后，通过 cmd 启动命令行窗口，输入 vertx version 命令，检查 Vert.x 安装是否成功。如果安装成功，会显示 Vert.x 的版本号和发布日期，如图 1-9 所示。

为测试 Vert.x 的编程，我们以最简单地显示“你好”程序，作为第一个 Vert.x 的应用案例，并使用 JavaScript 编写，当然也可以使用其他支持的语言编写。

在硬盘任何盘符下创建目录，如 E:\vertxapp，使用记事本编写如下示意代码，并保存为 app.js，这里使用 JavaScript 语言，每条代码的功能会在本书中详细介绍，读者在此了解即可。

```
var console = require('vertx/console');
console.log("你好");
```

图 1-8　Vert. x 启动目录 bin 的环境变量 path 配置

图 1-9　测试 Vert. x 安装是否成功

在命令行窗口进入到 E:\vertxapp 目录下，输入 vertx run app. js，即可启动 app. js 的运行，在控制台上输出“你好”，如图 1-10 所示。

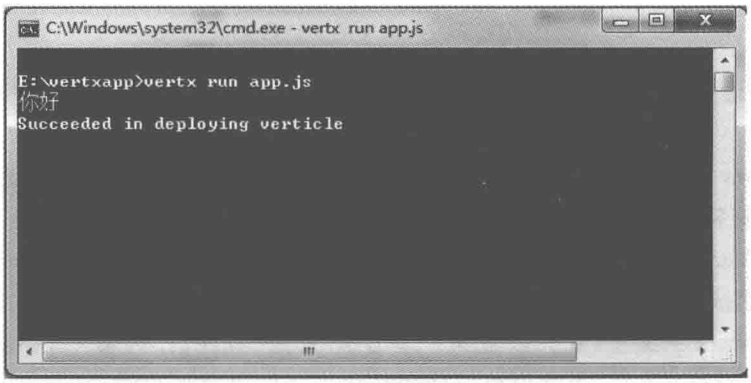

图 1-10　Vert. x 案例启动运行及控制台输出结果显示

但程序并没有终止运行，并直接返回到命令行提示符，这是由于 Vert. x 工作在无限的事件等待循环中。Vert. x 核心引擎循环等待新的事件发生，并在事件发生后，启动对应的 Verticle 组件运行，然后再等待下个事件，永无停止，直到终止该核心进程。我们可以通过

Ctrl+C 终止 Vert. x 的运行。

1.4 Vert. x 的特性

Vert. x 是最新的企业级开发的平台框架，它旨在解决多年来传统的企业级平台如 Java EE、MS. NET、PHP 等无法处理大量高并发的连接请求问题，并在基于 Node. js 高性能的基础上，解决 Node. js 编程复杂的难题应运而生的。以下 Vert. x 平台的核心特性注定其将在未来移动应用和企业级应用中全面取代传统的服务器平台，成为软件市场的佼佼者。

1. 多样性(Polyglot)

多样性是指 Vert. x 同时支持多种语言编程，目前支持 Java、JavaScript、CoffeeScript、Ruby、Python or Groovy，未来新版本将支持 Scala 和 Clojure。在一个 Vert. x 的应用项目中，可以混合使用以上语言编写 Vert. x 的应用组件，这些使用不同语言编写的组件可以相互调用，这在其他应用平台是无法实现的。例如 Node. js 只支持 JavaScript，Java EE 服务器平台只支持 Java，PHP 平台只支持 PHP 语言。

任何语言都不是万能的，每种语言都有自己的强项和特性，都有自己特别适合的应用领域。在编写一个企业级应用项目中，不同的业务处理采用最适合的语言编程，这个程序开发人员的梦想目标，现在由 Vert. x 实现了。

2. 简单性(Simplicity)

Vert. x 采用单一组件结构设计，即 Verticle，所有业务功能都使用 Verticle 这种单一的组件编程完成，克服以往应用框架和平台包含众多类型的组件模式，使得开发人员能极快适应 Vert. x 编程，加快项目的开发速度。

Vert. x 中所有的 Verticle 组件都是完全解耦合的，任何组件之间不能直接调用，只能通过在 Vert. x 的事件总线上发送事件来完成，彻底解决了传统应用系统中管理组件间相互依赖的复杂性，最终使得 Vert. x 应用编程极其简单高效。

Vert. x 使用单线程事件驱动的异步工作模式，编写 Vert. x 组件时，不需要考虑复杂的多线程编程难题，并不需要关注线程之间的调用、同步、加锁等繁琐处理编程，简化了编程代码，提高了编程效率。

3. 可扩展性(Scalability)

Vert. X 的设计是基于 Actor 模型的，你可以将一个或多个的程序放入到容器中去执行，可以很轻松地进行部署和升级，很方便地对应用进行水平扩展，动态升级。

4. 并发性(Concurrency)

在处理多用户并发连接请求时，Vert. x 摒弃了传统的多线程、同步工作、阻塞模式，而采用简单的单线程、异步工作、非阻塞模式，通过单线程内的事件驱动实现并发处理。

在多线程模式下，每个客户连接请求，服务器都会启动一个新的线程来处理此客户的请求处理。服务器的 CPU 内核是有限的，通常是双核、4 核及 8 核，要支持多于 CPU 内核的线程处理，需要操作系统采用某种调度机制进行线程的切换，而每个线程内都需要保存自己的变量和数据，要求在切换处理线程时，需要保存和还原原来的工作场景，这些都会浪费巨大的 CPU 和内存资源。

当线程过多时，导致处理性能急剧下降，因此多线程模式的平台无法支持大量并发连接