前言

缘起

自从一年半之前看到许式伟的博客，我认识了 Go 这一门语言，发现 Go 是 C 系的，个人又偏爱 C 语言，所以就开始了 Go 语言的学习之路，用三天时间学习了 Go 语言的所有语法和基础知识。恰逢当时手上有一些小项目练手，在项目开发中进一步发现 Go 语言具有三大优点：第一，性能好，我的 Mac 能够跑2万左右的 并发；第二，语法简单，对于以前有 C 语言基础的人来说非常容易上手，我仅用一天时间就熟悉了基本语法，Go 语言是一个上手即用的语言；第三，开发效率高， 目前有很多编辑器支持 Go 语言，对于开发效率有很大的提升，一般的小项目半天就能解决。通过一年多来对 Go 语言项目的实战累积，我越来越觉得 Go 是一门工 程语言，而不是其他学院派。无论是开发、测试、部署、项目规模的扩展，或者是团队协作，Go 语言考虑都非常周到；而且其语法恰当好处，不多不少，够用就是 它的设计原则，所以 Go 语言非常适合项目的开发。

选择 Go 语言，还有一部分是缘于我的个人崇拜，Go 语言的作者不乏鼎鼎大名的牛人：Robert Griesemer、Rob Pike 和 Ken Thompson，他们曾设计 C 语言和 Unix 系统，后来隶属 Plan9团队。重要的是，在 Go 语言的完全开源中，很多名人都参与了进来，使得这个项目越 来越完善：Go1.1出来后，性能提升了 30-50%，而且 GC（垃圾回收机制）已经达到了非常高的水准。相信在开源社区和大牛的共同推动下，Go 语言会 茁壮成长。

Go Web

我以前是 PHP 开发者，有十年左右的 Web 开发经验，但在 Go 语言的显著优势下，逐渐走向了 Go 语言的开发之路。我发现 Go 语言虽然有很强大的网络编 程库，但是在 Web 编程方面没有详细的介绍，也缺少一些比较实用的库，所以结合先前的 Web 开发经验，以及 Go 语言本身的网络编程库，开始了这本书的创作 过程，希望更多同行能够加入到 Go 语言的开发行列。

这本书主要分三部分，第一部分是 Go 语言的基础语法，主要介绍了 Go 语言的一些语法特性、环境配置和开发工具。第二部分是 Web 开发，主要介绍了 Go Web 的基本原理、表单处理、数据库操作、Session 和 Cookie 处理、文本处理、Socket 编程、安全加密、国际化和本地化、错误处理和调试、 如何部署和维护等知识点，并且针对整个 Web 开发中需要用到的知识点，结合 Go 语言代码的原理进行了详细的介绍，针对 Go 语言在 Web 开发方面不存在的工 具，提供了详细的实现方式。第三部分是应用框架 beego，主要介绍了 beego 框架的设计、实现及应用。目前书中提到的一些功能都可以在我的 github 找到相应的代码，方便读者进行深入的研究。

这是一本关于 Web 的书，我觉得特别适合以下几种开发者：

如果你是 PHP 或者其他动态语言爱好者，Go 语言不一定能带给你很大的惊喜，因为原来的速度不是根本问题。但如果是类似 API 应用方面，使用 Go 语言之后，你会发现性能得到了一个量的提升，这本书中就有详细介绍 API 开发的实例。

如果你是 C 语言爱好者，强烈建议你学习和使用 Go 语言。Go 语言称为21世纪的 C 语言，它不仅可以调用 C 语言程序，又可以提供足够的便利；虽然速度上稍有牺牲，但无关大雅。大部分场景下，Go 语言都能带给你与 C 语言媲美的性能，对于某些确实性能关键的场合，我们也可以通过 cgo，让 Go 语言和 C 语言

搭 配使用。

如果你是 Java 爱好者，那么也建议你学习一下 Go 语言，因为 Java 能给你的，Go 语言能给得更好。

如果你是 C++爱好者，那么赶紧来看看 Go 语言吧，因为光学习 C++特性的时间，已经可以开发多个 Go 语言项目了。

致谢

Go 社区里的同仁们给了我很大的支持，如果没有他们的反馈和帮助，我就不能顺利完成本书。非常感谢四月份平民、Hong Ruiqi、BianJiang、Oling Cat、Wenlei Wu、polaris、雨痕等网友的热心指导，还有很多 github 中的贡献者，本书是在大家共同协作努力下才得以完成。

我还要感谢符隆美编辑对我的支持，当我才思枯竭、延期脱稿时，她经常鼓励我、开导我，使我在压力下完成此书并最终出版。

最后要感谢家庭对我的莫大支持，妻子刘玉娟帮我收集资料，帮忙完成了本书的大部分整理工作，儿子们倾听我的思路想法，没有他们，我也没有毅力完成写作，谢谢他们。

谢孟军

2013年4月于上海

序言

推荐序一

很高兴听到谢孟军的《Go Web 编程》要出版。当谢孟军找我写推荐序时，尽管工作非常繁忙，我还是一口应承下来了。原因很简单，作为国内首家完全采用 Go 语言开发的公司，七牛非常 乐意见到 Go 语言社区的繁荣。去年在 Google Trends 上 Golang 关键字的搜索指数，中国排在全球首位（比美国多3倍），这是整个中国 Go 语言社区共同努力的结果。

远在2007年第2届 ECUG 大会，我讲了《我为什么选择了 Erlang》的议题。其中提到了我对未来软件产业趋势的判断：

存储与计算向服务端转移

从“PC 单机”到“强悍的服务器+多元化的终端”（手机、PC、PDA、电视机顶盒、车载终端）

这个趋势判断对我职业生涯的影响非常重大。它促使我放弃了近10年的桌面开发经验(包括大学时期)，转向服务端开发。正如我在《我为什么选择了 Erlang》中建议的那样：

要么就不写程序，要么就写服务器端的程序

当然，你也可以去撰写移动终端设备上的代码，在 PC 平台上做开发的空间很小

于是，我开始了长达四、五年之久的服务端开发最佳实践的探索。直到有一天，我遇到了 Go 语言。

我从来不认为自己是一个预言师，但关注过我的人可能都知道，我在新浪微博、《Go 语言编程》一书中都非常高调地下了一个论断：Go 语言将超过 C 语言、Java，成为未来十年最流行的语言。

为什么我可以如此坚定地相信，选择 Go 语言不会有错，并且相信 Go 语言会

成为未来十年最流行的语言？除了 Go 语言的并发编程模型深得我心外，Go 语言的各种语法特性显得那么深思熟虑、卓绝不凡，其对软件系统架构的领悟，让我深觉无法望其项背，处处带给我惊喜。

Go 语言给我的第一个惊喜，是大道至简的设计哲学。

Go 语言是非常简约的语言。简约的意思是少而精，少就是指数级的多。Go 语言极力追求语言特性的最小化，如果某个语法特性只是少写几行代码，但对解决 实际问题的难度不会产生本质的影响，那么这样的语法特性就不会被加入。Go 语言更关心的是如何解决程序员开发上的心智负担。如何减少代码出错的机会，如何 更容易写出高品质的代码，是 Go 语言设计时极度关心的问题。

Go 语言追求显式表达。任何封装都是有漏洞的，最佳的表达方式就是用最直白的表达方式。所以也有人称 Go 语言为“所写即所得”的语言。

Go 语言也是非常追求自然（nature）的语言。Go 不只是提供极少的语言特性，并极力追求语言特性最自然的表达，也就是这些语法特性被设计成恰如多少人期望的那样，尽量避免争议。事实上 Go 语言的语法特性上的争议非常少，这些也让 Go 语言的入门门槛变得非常低。

Go 语言给我的第二个惊喜，是最对胃口的并行支持。

我对服务端开发的探索，始于 Erlang 语言，并且认为 Erlang 风格并发模型的精髓是轻量级进程模型。然而 Erlang 除了语言本身不容易被程序 员接受外，其基于进程邮箱做消息传递的并发编程模型也小有瑕疵。我曾经在 C++中实现了一个名为CERL 的网络库,刚开始在 C++中完全模仿 Erlang 风格的并发编程手法，然而在我拿 CERL 库做云存储服务的实践中，发现了该编程模型的问题所在并做了相应的调整，这就是后来的 CERL 2.0版本。有意思的是，CERL 2.0与 Go 语言的并行编程思路不谋而合。某种程度上来说，这种默契也是我创办七牛时，Go 语言语法特性甚至都还没有完全稳定，我们技术选型就坚决地采纳 了 Go 语言的重要原因。

Go 语言给我的第三个惊喜，是 interface。

Go 语言的 interface，并非是你在 Java 和 C#中看到的 interface，尽管看起来有点像。Go 语言的 interface 是非侵入式的 接口，具体表现在实现一个接口不需要显式地进行声明。不过，让我意外的不是 Go 语言的非侵入式接口，非侵入式接口只是我接受 Go 语言的基础。在接口（或契 约）的表达上，我一直认为 Java 和 C#这些主流的静态类型语言都走错了方向。C++的模板尽管机制复杂，但是走在了正确的方向上。C++0x（后来的 C++11）呼声很高的 concept 提案被否，着实让不少人伤了心。但 Go 语言的 interface 远不是非侵入式接口那么简单，它是 Go 语言类型系统 的纲，这表现在：

1．只要某个类型实现了接口要的方法，那么我们说该类型实现了此接口。该类型的对象可赋值给该接口。

. 2．作为1的推论，任何 Go 语言的内置对象都可以赋值给空接口 interface{}。

3．支持接口查询。如果你曾经是 Windows 程序员，你会发现 COM 思想在 Go 语言中通过 interface 优雅呈现。并且 Go 语言吸收了其中最精华部分，而 COM 中对象生命周期管理的负担，却因为 Go 语言基于 GC（垃圾回收机制）方式的内存管理而不复存在。

Go 语言给我的第四个惊喜，是极度简化但完备的“面向对象编程（OOP）”

方法。

Go 语言废弃大量的 OOP 特性，如继承、构造/析构函数、虚函数、函数重载、默认参数等，简化的符号访问权限控制、将隐藏的 this 指针改为显式定义的 receiver 对象。Go 语言让我看到了 OOP 编程核心价值原来如此简单——只是多数人都无法看透。

Go 语言带给我的第五个惊喜，是它的错误处理规范。

Go 语言引入了内置的 error 类型及 defer 关键字来编写异常安全代码，让人拍案叫绝。下面这个例子，我在多个场合都提过。

```
f, err := os.Open(file)
if err != nil {
... // error processing
return
}
deferf.Close()
... // process file data
```

Go 语言带给我的第六个惊喜，是它功能的内聚。

一个最典型的案例是 Go 语言的组合功能。对于多数语言来说，组合只是形成复合类型的基本手段，这一点只要想想 C 语言的 struct 就清楚了。但 Go 语言引入了匿名组合的概念，它让其他语言原本需要引入继承这一新概念来完成事情，统一又到了组合这样的一个基础上。

在 C++中，你需要这样定义一个派生类。

```
class Foo : public Base {
...
};
```

在 Go 语言中你只要

```
type Foo struct {
Base
...
}
```

更有甚者，Go 语言的匿名组合允许组合一个指针。

```
type Foo struct {
*Base
...
}
```

这个功能可以实现 C++中一个无比晦涩难懂的特性，叫“虚拟继承”。但同样的问题，换从组合角度来表达，直达问题的本质，清晰易懂。

Go 语言带给我的第七个惊喜，是消除了堆与栈的边界。

在 Go 语言之前，程序员是清楚地知道哪些变量在栈上，哪些变量在堆上。堆与栈是基于现代计算机系统的基础工作模型上形成的概念，Go 语言屏蔽了变量定 义在堆还是栈上这样的物理结构，相当于封装了一个新的计算机工作模型。这一点看似与 Go 语言显式表达的设计哲学不太一致，但我个人认为这是一项了不起的工 作，而且与 Go 语言的显式表达并不矛盾。Go 语言强调的是对开发者的程序逻辑（语义）的显式表达，而非对计算机硬件结构的显示表达。对计算机硬件结构的高 度抽象，将更有助于 Go 语言适应未来计算机硬件发展的变化。

Go 语言带给我的第八个惊喜，是 Go 语言对 C 语言的支持。

可以这么说，Go 语言是除了 Objective-C、C++这两门以兼容 C 为基础目标的语言之外的所有语言中，对 C 语言支持最友善的一个。什么语言可 以直接嵌入 C 代码？没有，除了 Go 语言。什么语言可以无缝调用 C 函数？没有，除了 Go 语言。对 C 语言的完美支持，是 Go 语言快速崛起的关键支撑。还有比 C 语言更让人觊觎的社区财富么？那是一个取之不尽的金矿。

总而言之，Go 语言是一门非常具变革性的语言。尽管这四十多年来（从1970年 C 语言诞生开始算起）出现的语言非常之多，各有各的特色，让人眼花缭乱。但是我个人固执地认为，谈得上突破了 C 语言思想，将编程理念提高到一个新高度的，仅有 Go 语言而已。

Go 语言很简单，但是具备极强的表现力。从目前的状态来说，Go 语言主要关注服务器领域的开发，但这不会是 Go 语言的完整使命。

我们说 Go 语言适合服务端开发，仅仅是因为它的标准库支持方面，目前是向服务端开发倾斜：

网络库（包括 socket、http、rpc 等）

编码库（包括 json、xml、gob 等）

加密库（各种加密算法、摘要算法，极其全面）

Web（包括 template、html 支持）

而作为桌面开发的常规组件：GDI 和 UI 系统与事件处理，基本没有涉及。

尽管 Go 还很年轻，Go 语言1.0版本在2012年3月底发布，到现在才1年多，然而 Go 语言已经得到了非常普遍的认同。在国外，有人甚至提出“Go 语言将制霸云计算领域”。在国内，几乎所有你听到过名字的大公司（腾讯、阿里巴巴、京东、360、网易、新浪、金山、豆瓣等等），都有团队对 Go 做服务端 开发进行了小范围的实践。这是不能不说是一个奇迹。

与之相反的是，因为年轻，Go 语言的资料，尤其是中文资料极度匮乏。在这样的背景下，《Go Web 编程》这样一本有非常强的实践背景的图书出版了，这绝对是雪中送炭。

《Go Web 编程》围绕做一个 Web 服务相关的一个个问题域展开：表单处理、数据库、会话（Session）、安全、国际化和本地化、日志、部署与维护。最后，结合作者的实践，本书给出了一个参考的 Web 编程框架，以简化 Web 编程，提升开发效率。

无论是对那些只是听过 Go 语言而打算开始了解的朋友，还是对那些已经进行 Go 语言开发的朋友，本书都极具参考价值。

另外值得一提的是，除了《Go Web 编程》一书外，谢孟军也发起了 Go 语言标准库文档的翻译工作，这是一项艰苦的工作，但可以预期将对 Go 语言的发展起到重要作用，读者如果有意对开源贡献自己的一份力量，欢迎你能够积极参与其中。

七牛云存储 CEO 许式伟

2013年4月

推荐序二

很早就知道孟军兄在网上写一本关于 Go Web 编程的书，但是因为各种原因都没缘分仔细去看，最近因为工作原因，也开始接触并使用 Go 语言，才去看这本书，读完后，便觉得相见恨晚。

本书并不是 Go 语言的教程，只是在第一章和第二章介绍 Go 的运行开发环境

以及基本语法，但是受益于 Go 语言自身的简洁性，却也把 Go 语言的方方面面介绍得非常清楚。

然后介绍 Web 编程方面的 HTTP，Web Server，文本处理，Cookie，Session 等知识，同时提到了 Web 编程中的各种安全问题，比如 CSRF、XSS、Session 劫持、SQL 注入、密码安全等问题，并且给出了 Go 语言解决方案。

与后台数据库的交互是 Web 编程中非常重要的环节，本书不仅介绍了 MySQL，SQLite，PostgreSQL 等传统关系型数据库，同时对 MongoDB，Redis 这两位 NoSQL 阵营的明星产品也有涉及，但最值得一提的是，作者编写的开源 Go 语言 ORM 库。一提到 Web 编程，我们马上想 到的是 PHP、Python、Ruby 等动态语言以及基于这些语言的各种框架，如 PHP 阵营的 Zend Framework，Python 阵营的 Django，Ruby 阵营的 Ruby On Rails，诚然，动态语言的特性加速了我们的开发效率，但是框架带来的便利与高效才是至关重要的，这点我们从 Spring，Hibernate 等框架对 Java 社区的重要性就可以看出。其中 ORM 是框架中非常重要的一部分，它帮开发者隐藏了繁琐的 SQL 细节，非常轻松地完成数据库的增删改查。作者开源的 Go 语言的 ORM 库功能已经相对完整，算是我国 Go 语言社区里开源的精品之作了，能有效提高使用 Go 语言进行 Web 开发的效率，虽然也存在需要提高改进的 地方，但合抱之木生于毫末，九层之台起于累土，千里之行始于足下，只要坚持不懈，持续改进，未尝没有像 Spring 一样成为全球知名框架的可能。

本书的最后，还介绍了如何进行国际化与本地化的 Web 开发，讲解了如何调试、部署和维护方面的实践，提出了设计可扩展 Web 框架的建议。

本书以 Web 编程为主线，讲解了开发、测试、设计和部署等方面需要的知识，涵盖了一个 Web 站开发生命周期的方方面面，不仅是希望用 Go 语言开发 Web 服务的读者会受益匪浅，而且用其他语言的读者对 Web 编程的概念也会有清晰的认识。

Go 语言目标是成为集合解释型编程的轻松、动态类型语言的高效及静态类型语言的安全三大优点的编译型语言，同时它对网络编程与多核计算支持非常好。在 国内外，都已经有大型的 IT 公司在内部试水使用 Go 语言开发各种服务，其中也有不少成功案例。在技术社区，也有很多人开始宣传 Go，使用 Go，关注 Go， 相信在不久的将来，会有更多的人来使用 Go 语言来开发他们的 Web 服务，因为 Go 语言确实非常优秀而且实用。

京东商城云平台资深工程师，高级经理 郭理靖

2013年4月

媒体评论

七牛云存储 CEO 许式伟：为什么我可以如此坚定地相信，选择 Go 语言不会有错，并且相信 Go 语言会成为未来十年最流行的语言？除了 Go 语言的并发编程模型深得我心外，Go 语言的各种语法特性显得那么深思熟虑、卓绝不凡，其对软件系统架构的领悟，让我深觉无法望其项背，处处带给我惊喜。

京东商城云平台资深工程师，高级经理 郭理靖：Go 语言的目标是成为集合解释型编程的轻松、动态类型语言的高效及静态类型语言的安全三大优点的编译型语言，同时它对网络编程与多核计算支持非常好。

@waylau：作者是国内较早研究 GO 语言在网络应用的实践和推广者，该书讲述了 GO 语言在 Web 开发方面的应用，例子详实，文章充实。不管是新手还是老

鸟，读之受益匪浅。

@xgdapg：作者长期的 Web 开发经验和大量的实用技巧，使得本书的实用性远非其他空讲语法及概念的书籍可比。无论作为 Go 语言入门教程还是作为工具书，都是读者极好的选择。

@JessonChan：认真读过该书，作者讲解深入浅出，并且用 beego 和 beedb 开发了几个自己的项目，非常好，适合新人。

@aaronyue：作者作为国内 Go 语言的践行先驱，出手确实不一般。本书无论是学习 Go 语言还是 Web 编程相关概念和工具，都是一本值得反复阅读的好书。

@Xuyuanp：作为谷粉同是也是个码农，当然不能放过 Go 语言，这本书让我受益匪浅，非常感谢！