

Service Computing

Principle, Technology and Architecture for building effitive, elastic and solid services on cloud

开发 web 服务程序

1、概述

开发简单 web 服务程序 cloudgo，了解 web 服务器工作原理。

任务目标

1. 熟悉 go 服务器工作原理
2. 基于现有 web 库，编写一个简单 web 应用类似 cloudgo。
3. 使用 curl 工具访问 web 程序
4. 对 web 执行压力测试

相关知识

课件：<http://blog.csdn.net/pmlpml/article/details/78404838>

2、任务要求

基本要求

1. 编程 web 服务程序 类似 cloudgo 应用。
 - 要求有详细的注释
 - 是否使用框架、选哪个框架自己决定 请在 README.md 说明你决策的依据
2. 使用 curl 测试，将测试结果写入 README.md
3. 使用 ab 测试，将测试结果写入 README.md。并解释重要参数。

扩展要求

选择以下一个或多个任务，以博客的形式提交。

1. 选择 net/http 源码，通过源码分析、解释一些关键功能实现
2. 选择简单的库，如 mux 等，通过源码分析、解释它是如何实现扩展的原理，包括一些 golang 程序设计技巧。
3. 在 docker hub 申请账号，从 github 构建 cloudgo 的 docker 镜像，最后在 Amazon 云容器服务中部署。
 - 实现 Github - Travis CI - Docker hub - Amazon “不落地”云软件开发流水线

4. 其他 web 开发话题

tip: 源代码阅读

阅读源代码是学习 golang 绕不开的任务，否则你无法达到你期望的水平与能力。如何阅读源代码？这是一个非常复杂的话题，知识、经验和技巧都有很大作用。

X.1 准备工作

1. 了解对应语言的 Code Convention 非常重要，好的作品必须遵循这些开发约定，这些约定是实践中形成的。对于 Golang：
 - 建议首先阅读 [实效Go编程](#)。但这不是一个简单工作，但语言基本约定以及它的目录，你必须知道！
 - 使用 VSCode 作为编辑工具，它会提示变量命名、注释等基本要求
 - 了解程序目录的基本约定
 - 知道实体构造这类基本约定，如 `New Type()`, `type.New()`。它们在代码中非常常见
 - 总之，这是一个积累的过程！
2. 最好有一定的 OO 设计模式知识，这对正确理解优秀开源产品至关重要！
 - 面向对象的设计原则与设计模式知道的越多越好，有 java 基础最好
 - 建议读物：[java设计模式](#)，它是 java 库设计的经验总结。golang io 库就是完全模仿 java io 库，golang database 就是 java jdbc 的翻版，甲骨文告谷歌不是没道理的。
3. 了解你关注的产品的知识
 - 例如你计划了解 golang web 服务器的实现，必须会使用它，并了解基本使用
 - 必须去读官网文档，了解该内容的设计动机、设计理念
 - 在网上找相似文章，以加快速度

X.2 实现原理阅读

以 net/http 库 web 工作原理阅读为例：

1. 有原理图，分四个步骤：创建 ServerSocket，绑定并 listen，accept 连接，创建 go 线程服务一个连接。
2. 我们从入口函数 ListenAndServe 开始开始用 Ctrl 键开始追代码：
 - 关注函数、方法参数中的 接口 和 函数 参数，是接口一定要了解接口的定义。OO 设计原理与模式大概率从这里开始
 - 随时查阅 API 文档，了解相关类型的属性与方法
 - 忽视任何错误处理、分支处理。尽管其中有许多有趣的东西，也要放弃
 - 其中特别注意闭包、匿名函数、匿名类型这些编程技巧
 - 特别注意接口断言语法 `var.(type)`
 - 线程要注意上下文对象（context）的构建

X.3 实现细节阅读

以 net/http 库 DefaultServeMux 实现为例

1. 追到类型 ServeMux。当然的知道它的任务是将“用户请求中 path 映射到 Handler”

- map -> (path/name?, handler)
- muxEntry: Handler 是接口， pattern?

2. 关键代码

- pathMatch 函数，你已经知道了，这就是 path == pattern 的简单匹配
- 在看看 ServeMux 方法的代码，基本就验证了你的想法

X.4 模仿阅读

以 spf13/cobra/command.go 为例：

1. 你已经知道了 command 实现原理，见 [面向对象设计思想与 golang 编程](#) 最后图
2. 必须知道 cli 处理参数基本流程
3. 就是找设计模式要求的元素。
 - 以 Execute() 为中心，研究 setPara, parse, run 的实现
 - 子cmd存储、parent
 - 用简单实际命令，研究它在树结构command实例中执行的过程

利用它的代码，重写它的基本实现，可顺利完成！

X.5 测试自己

注意循序渐进，避免开始就搞复杂的东西。

分析 gorilla/mux （属于 X3）

1. 官网阅读它的功能与使用
2. 从源代码角度对比 DefaultServeMux 与 gorilla/mux
3. 有哪些收获？

Service Computing maintained by [pmlpml](#)

本站总访问量次，本站访客数人次，本文总阅读量次

Published with [GitHub Pages](#)