QINIU
七牛云存储

Golang用武之地

@许式伟

个人介绍

QINIU
七牛云存储

- 七牛云存储 CEO
- Go 语言大中华区首席布道师（自封）
  - 《Go 语言编程》作者（之一）
  - 《Programming in Go》译者（之一）
- 盛大祥云计划发起人
- 前金山技术总监
- WPS Office 2005 首席架构师

contents

- Golang的思维方式
- Golang的主战场
- Golang服务器的编程范式

Golang 思维方式

- 最小心智负担原则
  - 最小特性
  - 最少惊异
  - 最少犯错机会

Golang 思维方式

QINIU 七牛云存储

- Go, Next Java? No, Next C!
  - 少就是指数级的多
    - 最少特性原则：如果一个功能不对解决任何问题有显著价值，那么就不提供
    - 显式表达：所写即所得的语言
  - 最对胃口的并行支持
  - 类型系统的纲：interface
  - 极度简化但完备的OOP
    - struct 可以定义成员方法(method)，这是Go对OOP支持的所有内容
    - 简化的符号访问权限控制、显式的 this 指针
  - 错误处理规范
    - 函数多返回值、内置 error 类型、defer
  - 功能内聚：例如，强大的组合能力
  - 消除了堆与栈的边界消除了堆与栈的边界
  - 最友善的 C 语言的支持
  - http://open.qiniudn.com/go-next-c.pdf

Golang 思维方式

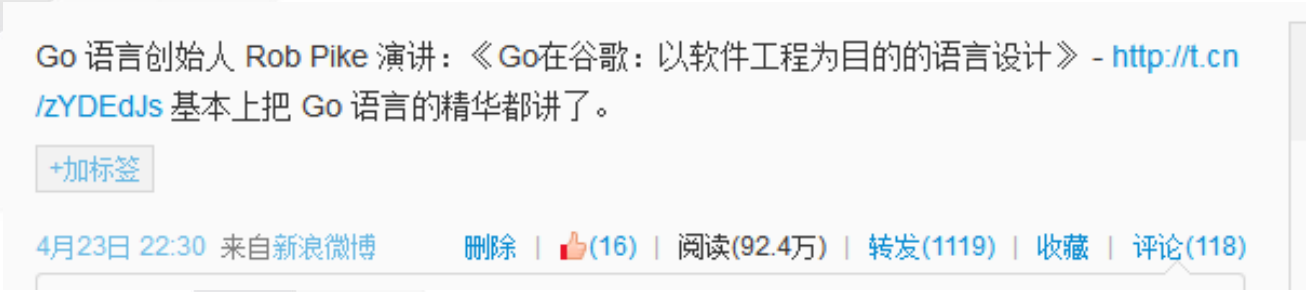
QINIU 七牛云存储

- Go, 基于连接与组合的语言
  - Pipeline 与并行模型
    - 在 Go 中实施 Pipeline 非常容易
    - 在 Go 中让任务并行化非常容易
  - 连接
    - Go 组件的连接是松散耦合的。彼此之间有最自然的独立性
    - Go 组件间的协议由 interface 描述，并在编译期进行 check
  - 组合
    - 不支持继承，却胜过继承
    - 不是 COM，但更胜 COM
  - http://open.qiniudn.com/thinking-in-go.mp4

Golang 思维方式

QINIU 七牛云存储

- 以软件工程为目的的语言设计
  - 快速编译
  - 严格的依赖管理
  - 代码风格的强一致性
  - 偏向组合而不是继承
  - http://www.weibo.com/1701886454/ztwNC2uj1



contents

- Golang的思维方式
- Golang的主战场
- Golang服务器的编程范式

今天讲什么？

QINIU 七牛云存储

- 不讲库
- 不通盘介绍Golang特性
- 谈谈
  - 为什么Golang是最好的服务端开发语言？

Golang的主战场

- 当前
  - 服务端开发
- 未来
  - 蚕食其他开发领域
    - 移动端/桌面/嵌入式/...

观点

- C 语言因为 Unix 的兴起而流行
- Go 语言因为云计算的兴起而流行

话外：什么是云计算？

QINIU 七牛云存储

- 云计算的实质是：服务外包取代技术外包
  - 技术挑战越来越大
    - 联网人数越来越多
    - 人均上网时间越来愈大
    - 富媒体越来越替代文字成为主流表达方式
      - 比如：微信的流行
    - 服务器的访问、存储压力越来愈大，把服务做好不容易
  - 运维难度越来越大
    - 服务越来越复杂，就算有相应的开源软件，看管软件的运行过程，以保证服务的健康运行，也已经是巨大的负担
  - 竞争越来越激烈
    - 巨头横行
    - 大量的同质化产品
    - 如何让自己跑得比别人快？创业者需要善假于物。
  - http://open.qiniudn.com/golang-and-cloud-storage.pdf

产业趋势

QINIU 七牛云存储

- 强悍的服务器 + 多元化的终端
  - 存储与计算向服务端转移
  - 终端多元化发展
    - 手机、PC、平板、电视/机顶盒、手表、眼镜、路由器
    - …
  - 2007年
    - http://open.qiniudn.com/why-i-choose-erlang.pdf

contents

- Golang的思维方式
- Golang的主战场
- Golang服务器的编程范式

服务器核心话题

QINIU 七牛云存储

- 高并发
  - 如何提高 IO 并发
  - 如何降低锁粒度

服务器编程范式

QINIU 七牛云存储

- 异步回调模型
  - 要提高 IO 并发，需要让 IO 完全异步化，等 IO 完成后发消息通知以完成后续业务
  - 代表：Node.js
- 轻量级进程模型
  - 要提高 IO 并发，需要让更多人（进程）一起干活
  - 代表：Golang
- 混合型
  - 介于轻量级进程模型与异步回调之间
    - 一般情形下倾向于轻量级进程模型，但是特定场合建议发异步消息
  - 代表：Erlang

线程间通讯：互斥

QINIU 七牛云存储

- 互斥 ==> 锁
  - 有共享变量，就需要锁！
  - Golang 服务器里，共享变量几乎是必然的
    - Why？
      - 因为服务器本身同时在响应很多请求。
      - 服务器本身就是共享资源（里面定义的变量被很多线程同时访问）。
    - Erlang 为什么没有锁？
      - 不是因为 Erlang 是 FP 语言，Erlang 没有“变量”。
        - » Why？为什么我说即便变量不可变，仍然无法避免锁？
      - 正确的原因是：Erlang 强制让所有请求串行化处理，Erlang 的服务器并不真正并行！
      - 如果某个 Golang 的服务器框架让所有请求串行化处理，那么也一样是不需要锁的。
      - Golang 服务器 runtime.GOMAXPROCS(1) 将程序设为单线程后，是否可以不需要锁？
        - » 不，仍然需要锁！Why？（请理解下goroutine工作原理）
        - » 单线程 != 所有请求串行化处理

Erlang 的服务器框架是个半吊子

QINIU 七牛云存储

- 混合型
  - 无法从一个方向贯彻到底
  - 确实避开了锁
    - 但是避得很纠结！
  - 详细：关于 Erlang 编程误区的讨论
    - http://weibo.com/1701886454/AeJIx4tm6?type=repost

Golang 服务器编程思想

QINIU 七牛云存储

- 纯正的轻量级进程模型！
  - 所有 API 操作都基于同步语言！
  - 要异步，拉个人帮我干活！
- 对于锁的态度：
  - 避无可避，无需逃避！

线程间通讯：互斥

QINIU 七牛云存储

- 锁的问题在哪？
  - 最大问题：不易控制
    - 锁 lock 但忘记 unlock 的结果是灾难性的，因为服务器相当于挂了（所有和该锁有关的代码都不能被执行）！
  - 次要问题：性能杀手
    - 锁会导致代码串行化执行
    - 但别误会：锁并不特别慢
      - 比线程间通讯其他原语（同步、收发消息）要快很多！
        - » 比锁快的东西：无锁、原子操作（比锁并不快太多）
      - 网上有人用 golang 的 channel 来实现锁，这很不正确

锁的最佳编程实践

QINIU 七牛云存储

- 善用 defer 和滥用 defer
  - 善用 defer 可以大大降低用锁的心智负担
  - 滥用 defer 可能会导致锁粒度过大
- 控制锁粒度
  - 不要在锁里面执行费时操作
    - 会阻塞服务器，导致其他请求不能及时被响应

锁的最佳编程实践

QINIU 七牛云存储

- 读写锁：sync.RWMutex
  - 如果一个共享资源(不一定是一个变量，可能是一组变量)，绝大部分情况下是读操作，偶然有写操作，则非常适合用读写锁。

- 锁数组：[]sync.Mutex
  - 如果一个共享资源，有很强的分区特征，则非常适合用锁数组
  - 比如一个网盘服务，网盘不同用户之间的资源彼此完全不相干

```
var mutexs [N]sync.Mutex

mutex := &mutexs[uid % N] // 根据用户id选择锁
mutex.Lock()
defer mutex.Unlock()
...
```

Q & A

QINIU
七牛云存储

@七牛云存储

@许式伟