

# Go语言发展状况

许式伟

2015-10-16

# Geekbang>

极客邦科技

全球领先的技术人学习和交流平台

扫我，码上开启新世界



# Geekbang>

InfoQ | EGO NETWORKS | StuQ

## InfoQ

专注中高端技术人员  
的社区媒体

## EGO NETWORKS

EXTRA GEEKS' ORGANIZATION  
高端技术人员  
学习型社交网络

## StuQ

实践驱动的IT职业  
学习和服务平台



促进软件开发领域知识与创新的传播



# 实践第一 案例为主

时间：2015年12月18-19日 / 地点：北京·国际会议中心

欢迎您参加ArchSummit北京2015, 技术因你而不同



ArchSummit北京二维码



【北京站】

2016年04月21日-23日



关注InfoQ官方信息  
及时获取QCon演讲视频信息

# 大纲

- Go语言语法及标准库变化
- Go语言未来发展的判断
- Go语言开源项目

# Go1.1

- 语言变化

- 支持方法值(Method values)

- p.Write 等价于:

- ```
func(b []byte) (int, error) {  
    return p.Write(b)  
}
```

- 区别于方法表达式(Method expressions):

- ```
(*Foo).Write, 结果类型为 func(p *Foo, b []byte) (int, error)
```

- int, uint 类型长度调整

- 自 Go1.1 后, int, uint 的尺寸统一是 64bits, 即使是在 32bits 平台下。

- go1.1 编译条件

- 在 go1.1 或以上的版本 编译:

- ```
// +build go1.1
```

# Go1.1

- 性能
  - 更多内联(inline): append, 接口类型转换
  - 全新实现 map, 更小的内存占用和CPU开销
  - GC 性能优化
  - 网络库性能优化: 减少上下文切换
- 标准库/工具
  - 一些尚未稳定的库从标准库移除
    - 迁移到 [code.google.com/p/go.exp](http://code.google.com/p/go.exp) 或 [code.google.com/p/go.text](http://code.google.com/p/go.text) 中
  - 新增的包(package)
    - go/format
    - net/http/cookiejar
    - runtime/race
- 详细信息
  - <http://tip.golang.org/doc/go1.1>

# Go1.2

- 语法变化
  - 三下标切片(Three-index slices): `array[begin:end:endcap]`
  - `cgo` 支持 C++
- 性能
  - 标准库 `compress/bzip2`, `crypto/des`, `encoding/json` 性能优化
- 标准库/工具
  - 新增的包(package): `encoding`, `image/color/palette`
  - `godoc`, `vet` 工具从 `go` 发布中移出, 迁移到
    - [code.google.com/p/go.tools/cmd/godoc](http://code.google.com/p/go.tools/cmd/godoc)
    - [code.google.com/p/go.tools/cmd/vet](http://code.google.com/p/go.tools/cmd/vet)
  - 测试覆盖率
    - [code.google.com/p/go.tools/cmd/cover](http://code.google.com/p/go.tools/cmd/cover)
- 详细信息
  - <http://tip.golang.org/doc/go1.2>

# Go1.3

- 语法变化：无
- 实现细节
  - GC
    - Go1.3以前，垃圾回收对指针的判断是经验性的：如果某个整数值恰巧是一个合法的指针，那么它会被当做指针处理
    - Go1.3调整为精确GC，这会导致以前的有些编程手法失效：
      - 如果有人意图把指针存储到uintptr而不是unsafe.Pointer，那么这块内存无法保证不被回收
      - 使用unsafe.Pointer存储整数也不会是一个好主意，因为这会干扰GC对真相的理解
  - map的遍历次序
    - Go1标准指出：对map的多次遍历不应该假设有同样的次序
    - 为了避免这样的假设发生，Go1.0遍历map选择起点时用了随机数
    - Go1.1用重新实现map后，小map的遍历没有引入随机数；Go1.3重新引入，以消灭此类潜在bug

# Go1.3

- 实现细节
  - cgo 变化: **C类型不再允许跨包访问**
    - 例如, 如果某个函数有 \*C.FILE 参数, 那么这个函数无法在其他包中使用(会编译失败)
      - 如果认为此代码是合理的, 可将其调整为 unsafe.Pointer
- 性能
  - 大大降低 defer 的内存开销
  - 引入并行 GC 算法, 大幅度改善 GC 效率
  - 竞争检测算法(race detector) 大幅提速
  - 优化 regexp 包, 改善大部分简单正则表达式的性能
- 标准库/工具
  - 新增的包(package): debug/plan9obj

# Go1.4

- 语法变化
  - 增加一种非常小概率使用的for range语法  
for range x { ... } 等价于 for \_ = range x { ... }
  - 不再自动对 \*\*T 进行解引用(dereference)
    - 对 var x \*\*T, 以前可直接用 x.Fn() 调用 T 的方法
      - 现在需要明确写 (\*x).Fn()
  - 引入 internal package 概念
    - 在 Go1.4 只对主路径(\$GOROOT)有效, 未来会全面应用到\$GOPATH
  - 引入 canonical import paths (权威导入路径) 概念
    - package pdf // import "rsc.io/pdf"
- 实现细节
  - 开始支持 Android (编译出binary或.so)
  - 运行时(runtime)
    - 大部分用Go重写(以前是C), 好处: 更加精确的GC
    - 放弃了分段栈(segmented stack)

# Go1.5

- 语法变化

- map 字面量简化写法

```
var x = map[Point]int{  
    {1.0, 2.1}: 1,  
    {2.3, 4.5}: 2,  
}
```

- 实现细节

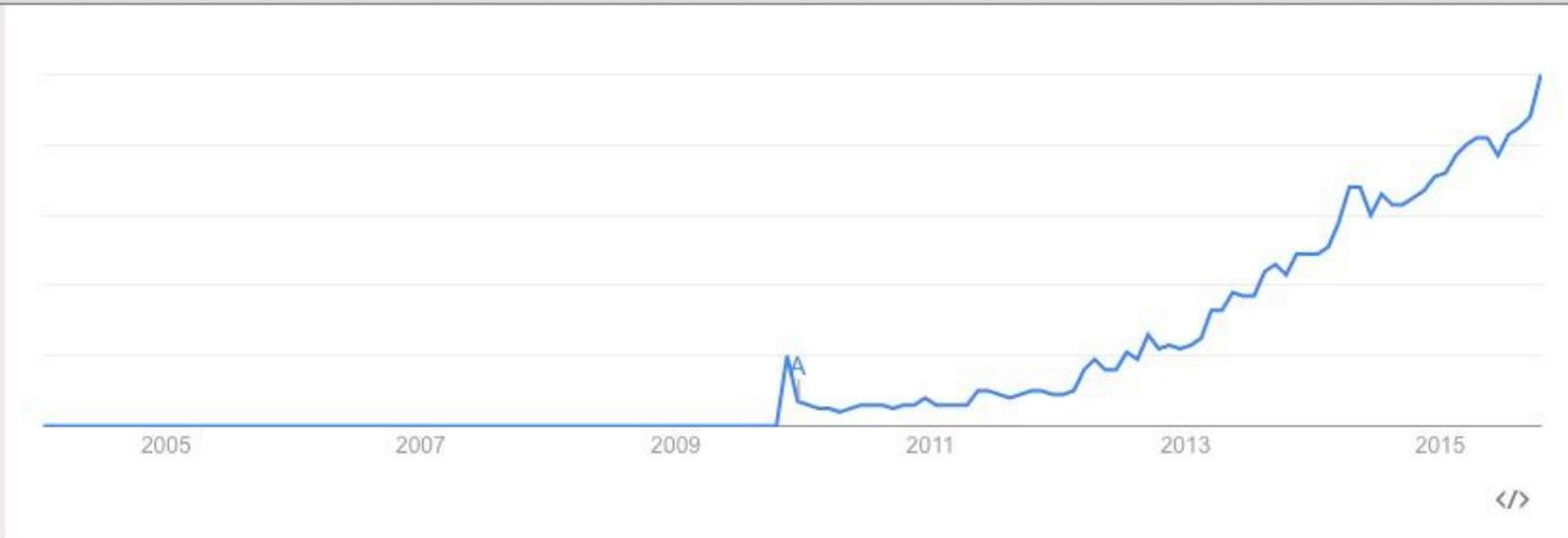
- 自举：不再有C，所有代码(包括编译器、链接器和运行时)都用Go和少量汇编实现
- GC：重写GC，进一步降低GC的latency
- GOMAXPROCS 默认是 CPU 数(以前是1)
- 开始支持 iOS
  - 为此 Go team 还提交了第一个 Go 写的 AppStore 应用 Ivy

# Go语言演进回顾

- 语法变化
  - 几乎没有增加什么显眼的语法特性，变化主要围绕：
    - 降低心智负担
    - 细节完善
    - 工程化
- Go本身的迭代的核心是实现优化，围绕：
  - 性能，性能，性能！
    - 性能优化，尤其是GC
  - 更多平台的移植
  - 更进一步的标准化，强化社区约定

# Go语言未来发展的判断

- 关于语言特性
  - Go语言3年发布了5个大版本，语言特性没有实质性的变化，是不思进取，还是深思熟虑？
    - 判断：别说3年，未来10年也不要指望Go语言特性有重大变化(泛型是唯一变数，但引入的可能性不是很大)。与其说是不思进取，还不如说是自信(对需求透彻理解后的强烈自信)。
- 关于Go语言的演进方向
  - 3到5年内，Go的主攻方向仍然是服务端开发
    - 比如内建 http 2.0 的支持 (更多的支持会以独立的开源库方式提供，未必直接包含在 Go 的发布包)
    - 尽管Go已经开始支持桌面端(Mobile)开发，但这仍然只是探索性和经验性的
  - 站在10年的维度，Go语言会在桌面端(Mobile)大放异彩，占据重要地位



### Regional interest ?



# Go的一些公开预测性言论

- Go语言将登上语言榜的首位，超过C和Java
  - 许式伟，2012年2月，新浪微博 & 《Go语言编程》序言
  - <http://weibo.com/1701886454/y7pROzJJj>
- Go语言将在两年内制霸云计算
  - 前VMware云平台CTO Derek Collison，2012年9月
  - <https://twitter.com/derecollison>
    - Prediction: Go will become the dominant language for systems work in IaaS, Orchestration, and PaaS in 24 months. #golang
- Go将统治下一个10年
  - DeferPanic 创始人 Ian Eyberg，2015年9月
  - <https://www.linkedin.com/pulse/go-dominate-next-decade-ian-eyberg>

# 话外：非技术人员眼中的Go...

deferPanic retweeted



**Jerry Chen** @jerrychen · Jul 8

Google's contribution of [@GoLang](#) could be one of the most lasting & influential open source contributions by any company. [@GopherCon](#)



7



10



Google开源的Go语言，将成为最持久、最有广泛影响力的开源系统之一 (投资人眼中)

“almost everything new and interesting is being written in Go” (BD眼中)



**Brian Flannery**

3rd

Software and IT Services Business Development, Marketing, and Alliances

Raleigh-Durham, North Carolina Area | Computer Software

# Go语言开源项目

- 云计算领域
  - 容器(及相关): runc, docker, coreos, codetainer, kitematic
  - 微服务(含自动化部署/服务发现/调度器/etc): flynn, tsuru, deis, otto, terraform, consul, skydock, nomad, kubernetes
  - 数据库(及相关): etcd (键值), boltdb (键值/事务), cayley(图), prometheus/seriesly (时间序列), cockroachdb, TiDB, influxdb, goleveldb, vitess, reborndb, pgweb, beedb
  - 其他: skydns, gryffin (Web安全), bosun (监控)
- 开发框架&工具
  - Web框架: utron, gin, revel, beego
  - 反向代理: traefik, v2ray
  - Git相关: git-lfs, gogs
  - 其他: sift (grep的替代品), gologin(授权), lantern, gotty, gowalker
- 开源应用
  - 同步型网盘: syncthing
  - 编辑器: lime

# 不是结论

- Go 发展得到底怎么样了？