Introduction to Go

rheno.manggala@uangteman.com

Agenda

- History
- What
- How to + ‘Hello World’
- Features + Demo
- Who is using go ?

History

- September 21, 2007. At Google
- Robert Griesemer, Rob Pike and Ken Thompson

What

- Based on C
- Compiled or Interpreted
- Type Safety
- Garbage Collected
- Efficient
- Lightweight Concurrency ?

How to

- Download at https://golang.org/dl
- Installation :
  1. brew install go
  2. Source (Linux and Unix)

‘Hello World’

```
package main

import "fmt"

func main() {

  fmt.Println("Hello Wolrd")

}
```

Features

Variables

- var <name-variable> <type-variable>
- <name-variable> := <initialization-value> (same as declaration with value)
- <name-variable> = <initialization-value> (change value name-variable)
- var <name-variable> <type-variable> = <initilize value>
- const <name-variable> = <value>

Variables (Continue)

- bool, string, int, int8, int16, int32, int64, uint, uint8, uint16, uint32, uint64, uintptr, byte (same as uint8), rune (int32, unicode), float32, float64, complex32, complex64

NO DEMO?
imgflip.com

Function

```go
package main

import "fmt"

func add(x int, y int) int {
    return x + y
}

func main() {
    fmt.Println(add(42, 13))
}
```

DEMO TIME ....
memegenerator.net

Looping

```
/* standard for */

for i := 0; i < 10; i++ {

          sum += i

}

/* another for */

for ; sum < 10 ; {

          sum += sum

}

/* while */

for sum < 10  {

     sum += sum

}
```

Condition

- if <condition-1> && <condition-2> {}
- if <variable> := <value> ; <condition> {}
- else if <condition-1> && <condition-2> {}
- switch <variable> { case <value> : <statement> }
- switch { case <condition> : <statement> }

ENOUGH TALK
SHOW ME A
DEMO!!!
memegenerator.net

Pointers

- Same like C
- <pointer> = &<variable>

Structs (Collection of Fields)

- Example :

```
type <name> struct {

 <variable-name> <variable-type>

}
```

- Example :

<name>.<variable-name>

38
DEMO TIME ....
memegenerator.net

Array vs Slice

- var <variable-name> [10] <variable-type>
- var <variable-name> [] <variable-type>

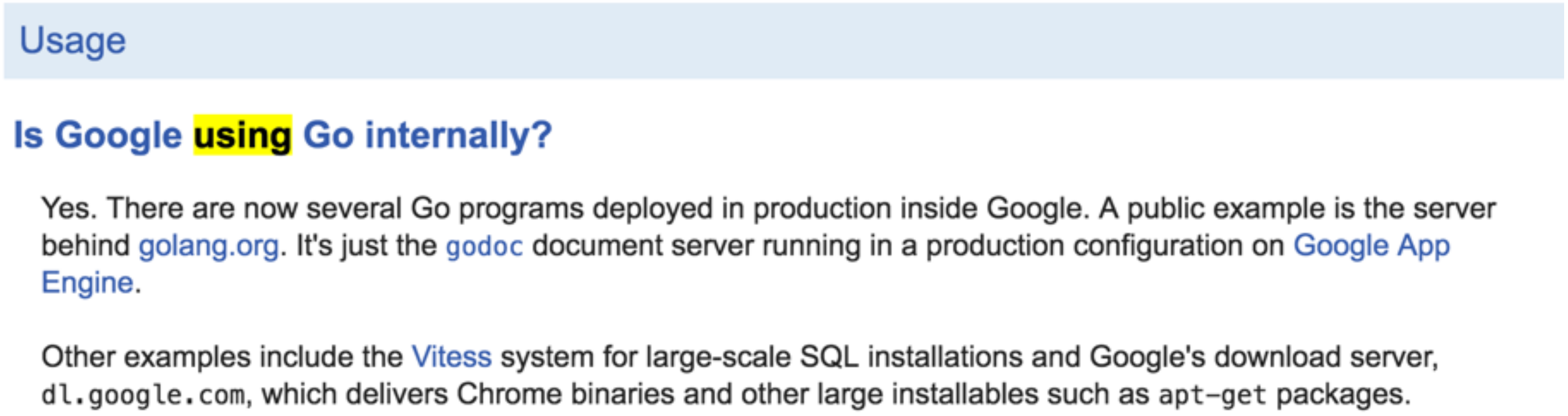
Map

- var m = make(map[<variable-type>] <variable-type>)

NO DEMO?
imgflip.com

Goroutine

Who is using go



Usage

Is Google using Go internally?

Yes. There are now several Go programs deployed in production inside Google. A public example is the server behind golang.org. It's just the `godoc` document server running in a production configuration on Google App Engine.

Other examples include the Vitess system for large-scale SQL installations and Google's download server, `dl.google.com`, which delivers Chrome binaries and other large installables such as `apt-get` packages.

Open Sourcing Our Go Libraries

Patrick Lee | July 1, 2014 | 34 comments 28 160 2

Dropbox owes a large share of its success to Python, a language that enabled us to iterate and develop quickly. However, as our infrastructure matures to support our ever growing user base, we started exploring ways to scale our systems in a more efficient manner. About a year ago, we decided to migrate our performance-critical backends from Python to Go to leverage better concurrency support and faster execution speed. This was a massive effort–around 200,000 lines of Go code–undertaken by a small team of engineers. At this point, we have successfully moved major parts of our infrastructure to Go.

One recurring theme that hindered our development progress was the lack of robust libraries needed for building large systems. This is not surprising since Go is still a very young language. To address this issue, our team started building various libraries to provide better abstractions, such as connection management and a memcache client. We are very excited to announce that we are open sourcing these libraries to help the broader community build large scale production systems.

Reference

- https://golang.org/doc/faq
- https://talks.golang.org/2012/splash.article
- https://blogs.dropbox.com/tech/2014/07/open-sourcing-our-go-libraries/

To be Continued