Let's Go !

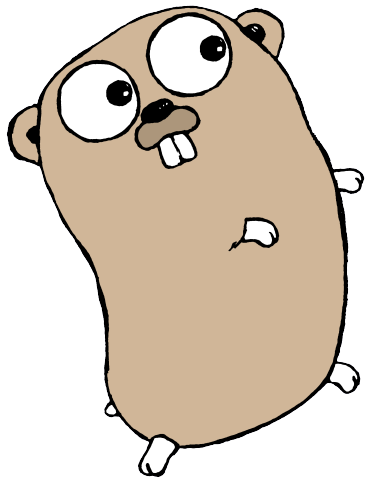
A brief introduction to Google's new language

Aurélien Dumez

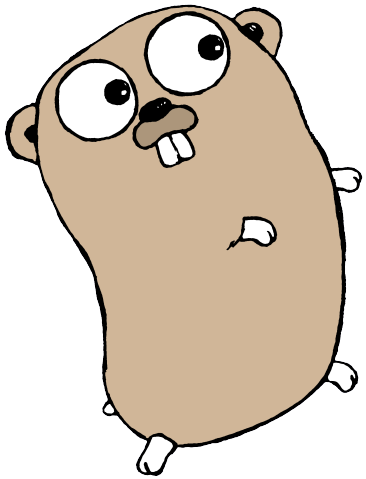
Inria Bordeaux - Sud-Ouest
aurelien.dumez@inria.fr

Tuesday, October 2nd 2012

Content - 1/2



Content - 2/2



Motivations

- **Web dev** : fast, efficient, secure, low requirements

- **PHP** : « a fractal of bad design »
- **Python** : too many solutions :
  django, flask, karrigell, nagare, pylons, pyramid, turbogears, web2py, web.py, zope... and Python 3 ?
- **Perl** : not my favourite language
- **Ruby** : (on Rails) last try (2008) was not conclusive
- **Java** : does not suit my needs

- **Other solutions** : what do you suggest ?

Motivations

- **Web dev** : fast, efficient, secure, low requirements

- **PHP** : « a fractal of bad design »
- **Python** : too many solutions :
  django, flask, karrigell, nagare, pylons, pyramid,
  turbogears, web2py, web.py, zope... and Python 3 ?
- **Perl** : not my favourite language
- **Ruby** : (on Rails) last try (2008) was not conclusive
- **Java** : does not suit my needs

- **Other solutions** : what do you suggest ?

Motivations

- **Web dev** : fast, efficient, secure, low requirements

- **PHP** : « a fractal of bad design »
- **Python** : too many solutions :
  django, flask, karrigell, nagare, pylons, pyramid, turbogears, web2py, web.py, zope... and Python 3 ?
- **Perl** : not my favourite language
- **Ruby** : (on Rails) last try (2008) was not conclusive
- **Java** : does not suit my needs

- **Other solutions** : what do you suggest ?

Motivations

- **Web dev** : fast, efficient, secure, low requirements

- **PHP** : « a fractal of bad design »
- **Python** : too many solutions :
  django, flask, karrigell, nagare, pylons, pyramid, turbogears, web2py, web.py, zope... and Python 3 ?
- **Perl** : not my favourite language
- **Ruby** : (on Rails) last try (2008) was not conclusive
- **Java** : does not suit my needs

- **Other solutions** : what do you suggest ?

History

Designers

- **Robert Griesemer** (V8 JS Engine, Java HotSpot...)
- **Rob Pike** (Bell Labs, Unix, Plan 9, Inferno, Limbo...)
- **Ken Thompson** (Bell Labs, Unix, Plan 9, B, UTF-8...)

Milestones

- Nov. 2009 : new public open source project
- 28-03-2012 : version 1
- 25-04-2012 : version 1.0.1
- 13-06-2012 : version 1.0.2

History

Designers

- **Robert Griesemer** (V8 JS Engine, Java HotSpot...)
- **Rob Pike** (Bell Labs, Unix, Plan 9, Inferno, Limbo...)
- **Ken Thompson** (Bell Labs, Unix, Plan 9, B, UTF-8...)

Milestones

- Nov. 2009 : new public open source project
- 28-03-2012 : version 1
- 25-04-2012 : version 1.0.1
- 13-06-2012 : version 1.0.2

History

Designers

- **Robert Griesemer** (V8 JS Engine, Java HotSpot...)
- **Rob Pike** (Bell Labs, Unix, Plan 9, Inferno, Limbo...)
- **Ken Thompson** (Bell Labs, Unix, Plan 9, B, UTF-8...)

Milestones

- Nov. 2009 : new public open source project
- 28-03-2012 : version 1
- 25-04-2012 : version 1.0.1
- 13-06-2012 : version 1.0.2

Principles

Principles

According to the Go Team

« Here at Google, we believe programming should be **fast**, **productive**, and most importantly, **fun**. That's why we're excited to **open source** an experimental new language called Go. Go combines the development speed of working in a **dynamic language like Python** with the **performance and safety of a compiled language like C** or C++. Typical builds feel instantaneous ; even large binaries compile in just a few seconds. And the compiled code runs **close to the speed of C**. Go lets you move fast. »

–Robert Griesemer, Rob Pike, Ken Thompson, Ian Taylor, Russ Cox, Jini Kim and Adam Langley –11 november 2009

Language - Characteristics

- Compiled language
- Strongly typed (no implicit type conversions)
- Low level (pointers)
- Garbage collector
- Not object-oriented
- No exception management
- Native concurrency support (goroutines and channels)
- Good standard library (Python's spirit)

- UTF-8 support, even in source files (var héhé int = 42)
- braces but implicite semicolons (vicious traps !)
- On the web, seek for « golang »

Language - Characteristics

- Compiled language
- Strongly typed (no implicit type conversions)
- Low level (pointers)
- Garbage collector
- Not object-oriented
- No exception management
- Native concurrency support (goroutines and channels)
- Good standard library (Python's spirit)

- UTF-8 support, even in source files (var héhé int = 42)
- braces but implicite semicolons (vicious traps !)
- On the web, seek for « golang »

Language - Characteristics

- Compiled language
- Strongly typed (no implicit type conversions)
- Low level (pointers)
- Garbage collector
- Not object-oriented
- No exception management
- Native concurrency support (goroutines and channels)
- Good standard library (Python's spirit)

- UTF-8 support, even in source files (var héhé int = 42)
- braces but implicite semicolons (vicious traps !)
- On the web, seek for « golang »

Language - SDK

- GNU/Linux, OS X, Windows and FreeBSD (and source)
- BSD license
- Unzip and start coding, no cumbersome install

- **go build** : run the compiler
- **go fmt** : standard code formatting
- **go doc** : browse the doc (web server available)
- **go install** : install a package
- **go get** : download and install a package
- **go test** : lightweight unit-testing

Language - SDK

- GNU/Linux, OS X, Windows and FreeBSD (and source)
- BSD license
- Unzip and start coding, no cumbersome install

- **go build** : run the compiler
- **go fmt** : standard code formatting
- **go doc** : browse the doc (web server available)
- **go install** : install a package
- **go get** : download and install a package
- **go test** : lightweight unit-testing

Language - SDK

- GNU/Linux, OS X, Windows and FreeBSD (and source)
- BSD license
- Unzip and start coding, no cumbersome install

- **go build** : run the compiler
- **go fmt** : standard code formatting
- **go doc** : browse the doc (web server available)
- **go install** : install a package
- **go get** : download and install a package
- **go test** : lightweight unit-testing

Language - vs

	Go	C
variables	`var i int`	`int i`
structures	`type st struct {i int` `c byte}`	`struct st { int i;` `char c; };`
loops	`for i:=1; i<=10; i++ {}`	`for (i=1; i<=10; i++) {}`
conditionals	`if i > 0 {} else {}`	`if (i>0) {} else {}`
functions	`func f(in int) int {}`	`int f(int in) {}`

Language - vs

	Go	C
variables	`var i int`	`int i`
structures	`type st struct {i int` `c byte}`	`struct st { int i;` `char c; };`
loops	`for i:=1; i<=10; i++ {}`	`for (i=1; i<=10; i++) {}`
conditionals	`if i > 0 {} else {}`	`if (i>0) {} else {}`
functions	`func f(in int) int {}`	`int f(int in) {}`

Language - example 1 of 1337

```
package main

import "fmt"

func swap(a int, b int) (int, int) {
        return b, a
}

func main() {
        fmt.Println(swap(1, 2))
}
```

Language - 1335 examples left

```
func cp(to, from string)
    (written int64, err os.Error) {

    src, err2 := os.Open(from, os.O_RDONLY, 0)
    if err2 != nil { return }
    defer src.Close()
    dst, err2 := os.Open(to, os.O_WRONLY|os.O_CREATE,
                                        0644)
    if err2 != nil { return }
    defer dst.Close()
    return io.Copy(dst, src)
}
```

Language - Examples - Interfaces

```
type T struct { i int }

func (p *T) Get() int { return p.i }
func (p *T) Put(v int) { p.i = v }

var t T

t.Put(2)

fmt.Println(t.Get())
```

Language - Examples - Web

```
package main

import ("net/http"
        "io"
        "code.google.com/p/go.net/websocket")

func EchoServer(ws *websocket.Conn) {
        io.Copy(ws, ws);
}

func main() {
        http.Handle("/echo", websocket.Handler(EchoServer));
        err := http.ListenAndServe(":12345", nil);
        if err != nil {
                panic(err)
        }
}
```

Language - Some packages

Archive/Compress

tar, zip, bzip2, gzip

Crypto/Hash

aes, elliptic, sha1, x509

Net/Web

http, html, cgi, smtp

Encoding

base64, csv, json, xml

Strings

regexp, strconv, unicode

System

os, syscall, syslog

Language - Some packages

Archive/Compress

tar, zip, bzip2, gzip

Crypto/Hash

aes, elliptic, sha1, x509

Net/Web

http, html, cgi, smtp

Encoding

base64, csv, json, xml

Strings

regexp, strconv, unicode

System

os, syscall, syslog

Gommunity

Official Website

www.golang.org

Go Projects

godashboard.appspot.com

Google Group

groups.google.com

Go Lang Community Wiki

code.google.com

Books

- Learning Go (CC BY-NC-SA)
- An Introduction to Programming in Go (CC BY 3.0)

Gommunity

Official Website

www.golang.org

Go Projects

godashboard.appspot.com

Google Group

groups.google.com

Go Lang Community Wiki

code.google.com

Books

- Learning Go (CC BY-NC-SA)
- An Introduction to Programming in Go (CC BY 3.0)

Success stories

- **Canonical** : backend infrastructure
- **BBC World News** : backend for different games
- **Argonne National Laboratory** : platform for computation, storage, and distribution of scientific data (Shock)

Success stories

- **Canonical** : backend infrastructure
- **BBC World News** : backend for different games
- **Argonne National Laboratory** : platform for computation, storage, and distribution of scientific data (Shock)

Conclusion

- Good language for system and network programming
- Young language :
  - How stable is the specification ?
  - Several libraries for one task, no library for another one
  - Enthusiastic community but many projects start and die quickly
- Go is used internally in Google : more precisely ?

Conclusion

- Good language for system and network programming
- Young language :
  - How stable is the specification ?
  - Several libraries for one task, no library for another one
  - Enthusiastic community but many projects start and die quickly
- Go is used internally in Google : more precisely ?

Thank you, any questions ?