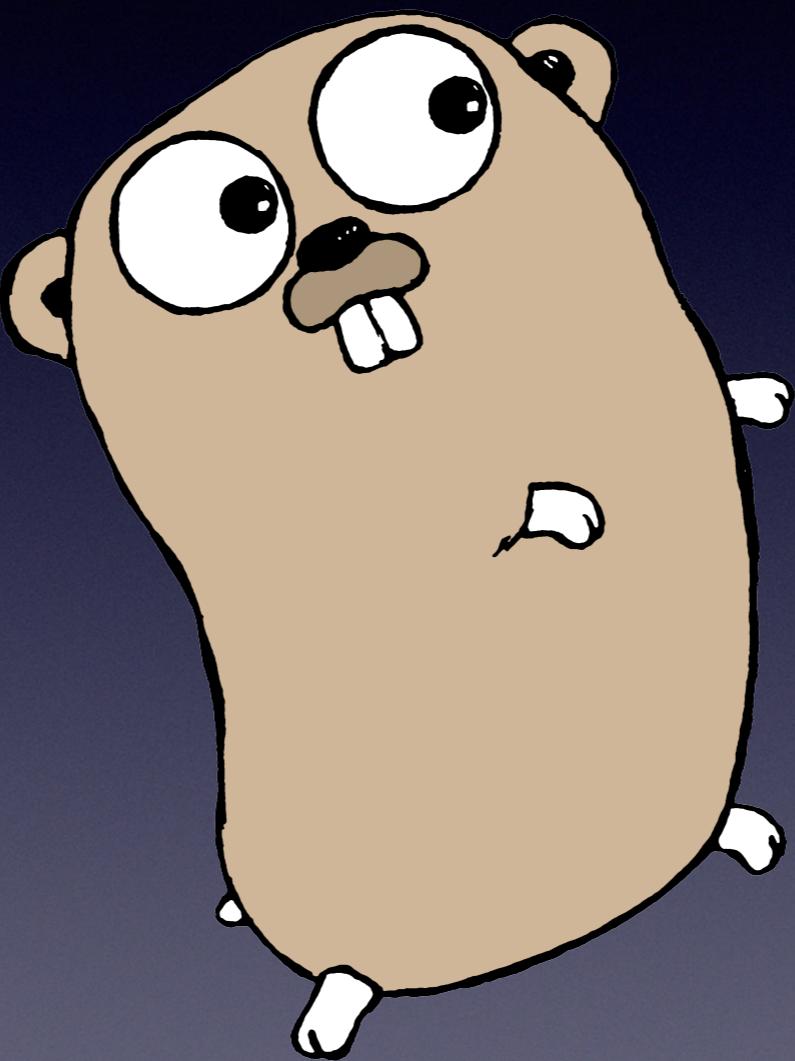


Go



Gordon Mickel
Open-Source Software Entwicklung
09. Dezember 2014

Kurze Geschichte von Go

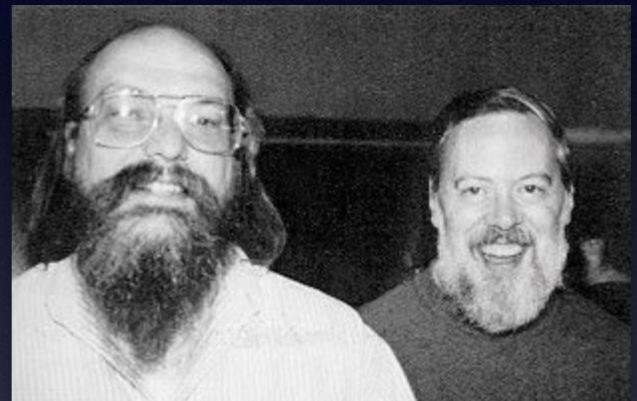
“The three of us got together and decided that we hated C++”



Rob Pike



Robert Griesemer



Ken Thompson

- 2007: Die Idee
- 2009: Go wird Open Source (BSD Lizenz)
- 2012: Version 1.0

Warum?

- Unzufriedenheit über bestehenden Sprache wie C++ und Java
- Systemprogrammierung, High performance Servers, skalierbare Netzwerkdienste, Cluster- und Cloud-Computing
- Soll dem Entwickler die Arbeit erleichtern

Schnellüberblick

- C-ähnliche Syntax
- Kompiliert
- Statisch typisiert (aber mit Typinferenz)
- Eingebaute Parallelisierung
- Schnell
- "OOP-ish"

Syntax

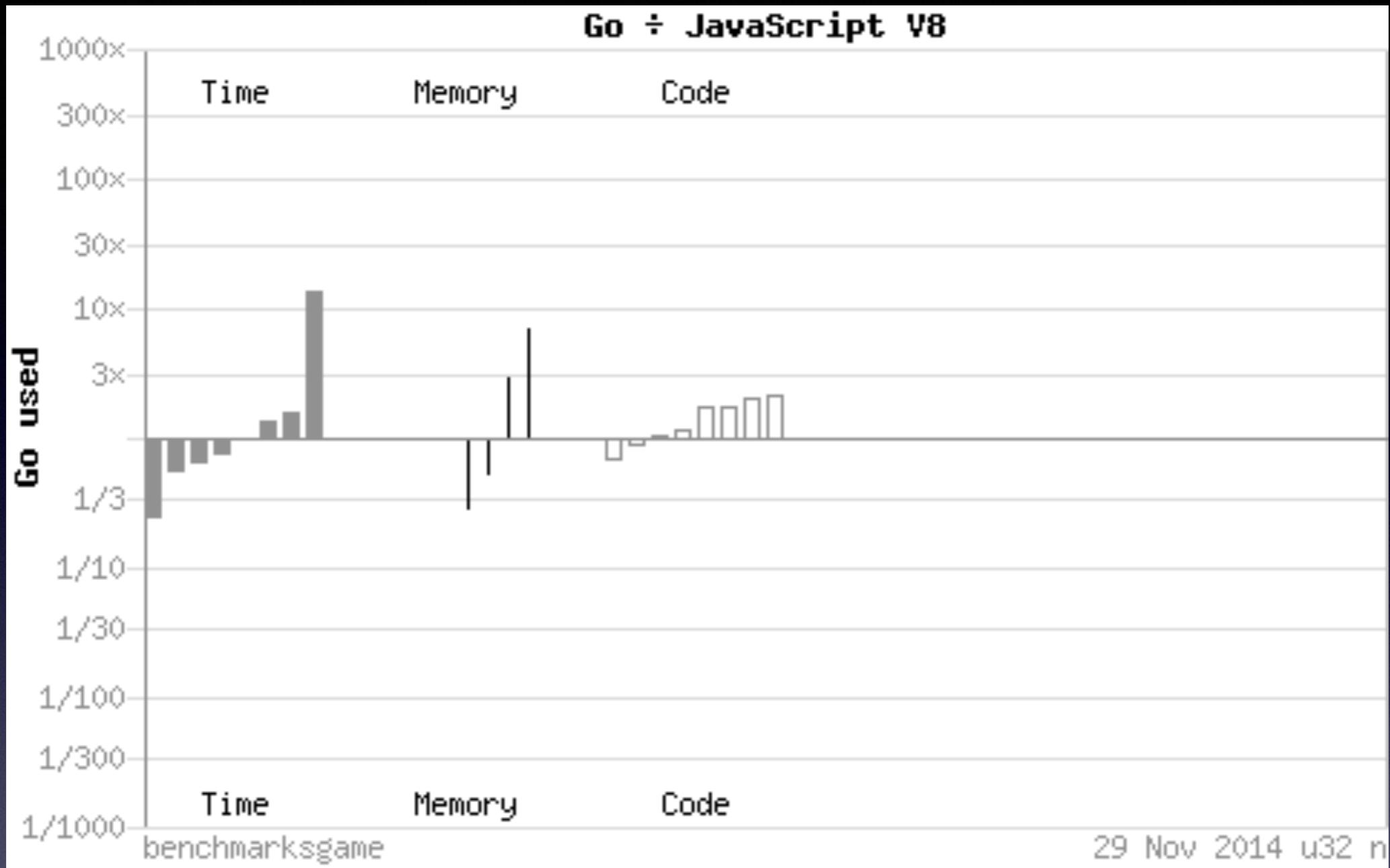
- Einfach und schnell zu lernen
- Wenig keywords, z.B. nur for Loops
- Mehrere return Werte
- Pointers, ohne Arithmetik

Go Playground

- Zum Testen und Austauschen von Go-Progammen

Geschwindigkeit

- Sehr schnelle Kompiliergeschwindigkeit
- Go soll schnell sein (ist es aber wirklich so?)
 - vergleichbar wie Java, schneller wie Python / Ruby



Go vs Javascript V8

Ökosystem

- Go get - ähnlich wie pip/gem/npm
- Funktioniert mit Github / Bitbucket / Google Code
- Gute Repository Suchmaschinen
- Keine gute Programmierumgebung (IDE)

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

Konventionen

- Convention over configuration
- Packages
 - Fest vordefinierte Verzeichnisstruktur
- Tests & Benchmarks
 - *_test.go

Standard Library

- Umfangreich
- Gut dokumentiert
- Beispiele
 - HTTP Web Server + Template library
 - Kompression, Verschlüsselung, JSON / XML / CSV
 - Profiling, debugging
 - Bildverarbeitung
 - Kein GUI Toolkit

Concurrency

- Goroutines und Channels
- Goroutines sind microthreads mit einem internen Scheduler
- Channels sind synchronisierte Kommunikationskanäle
- Non-blocking I/O nicht nötig

C Bindings

- cgo
- C Libraries trivial einzubinden
- Go Code von C aufrufen auch einfach möglich

Vielen Dank!