Intro to Go

Jan Urbansky

What is Go?

- Created at Google in 2009
- Version 1.0 in 2012
- Statically Typed
- Compiled (Cross compilation)
- Binaries (No VM or Engine needed)

Language Features

- C-like syntax

- Automatic type inference

- Structs as „Classes“

- Structs have Methods

- Functions have multiple return values

- Implicit Interfaces

- Closures, defer

- No exceptions

- Arrays, Slices, Maps (no generics)

- Packages as Namespaces

Goroutines

- Goroutines are the concurrency of go
- They are functions that run in the background
- Comparable to threads but more lightweight
- GR are mapped to OS threads
- If all GR wait (deadlock), the program exits
- Communication between GR is possible with shared Variables or Channels
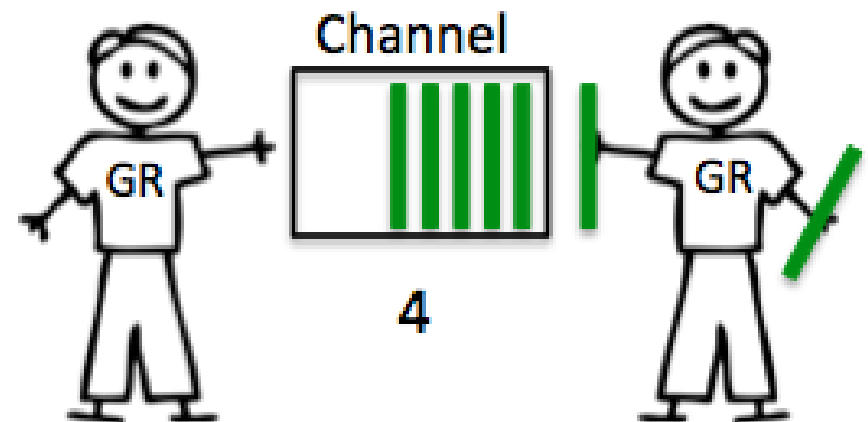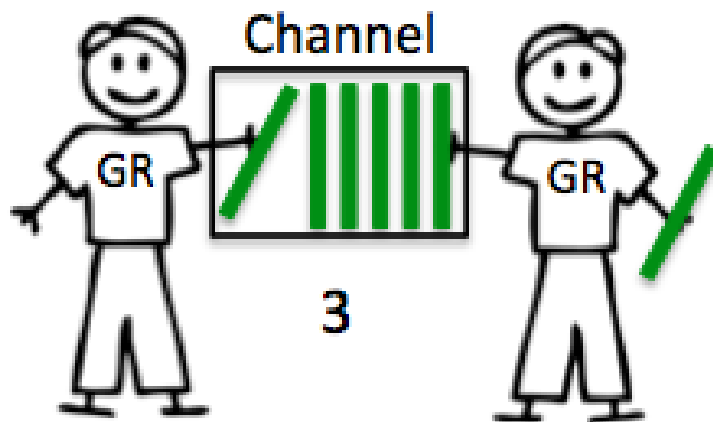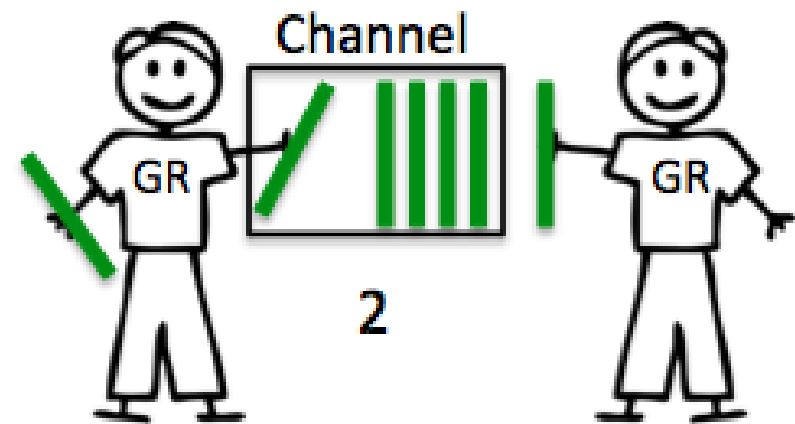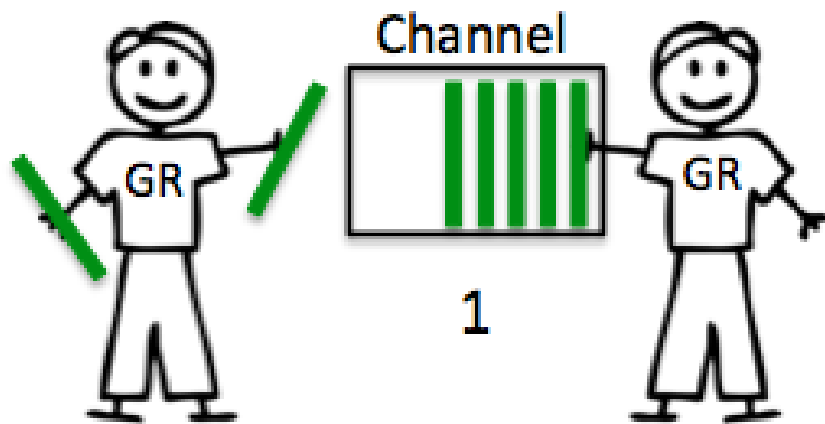- Shared Variables → Need for mutex

Channels

- Type safe message queue

- Buffered or Unbuffered

- Read:
  x := ← myChan
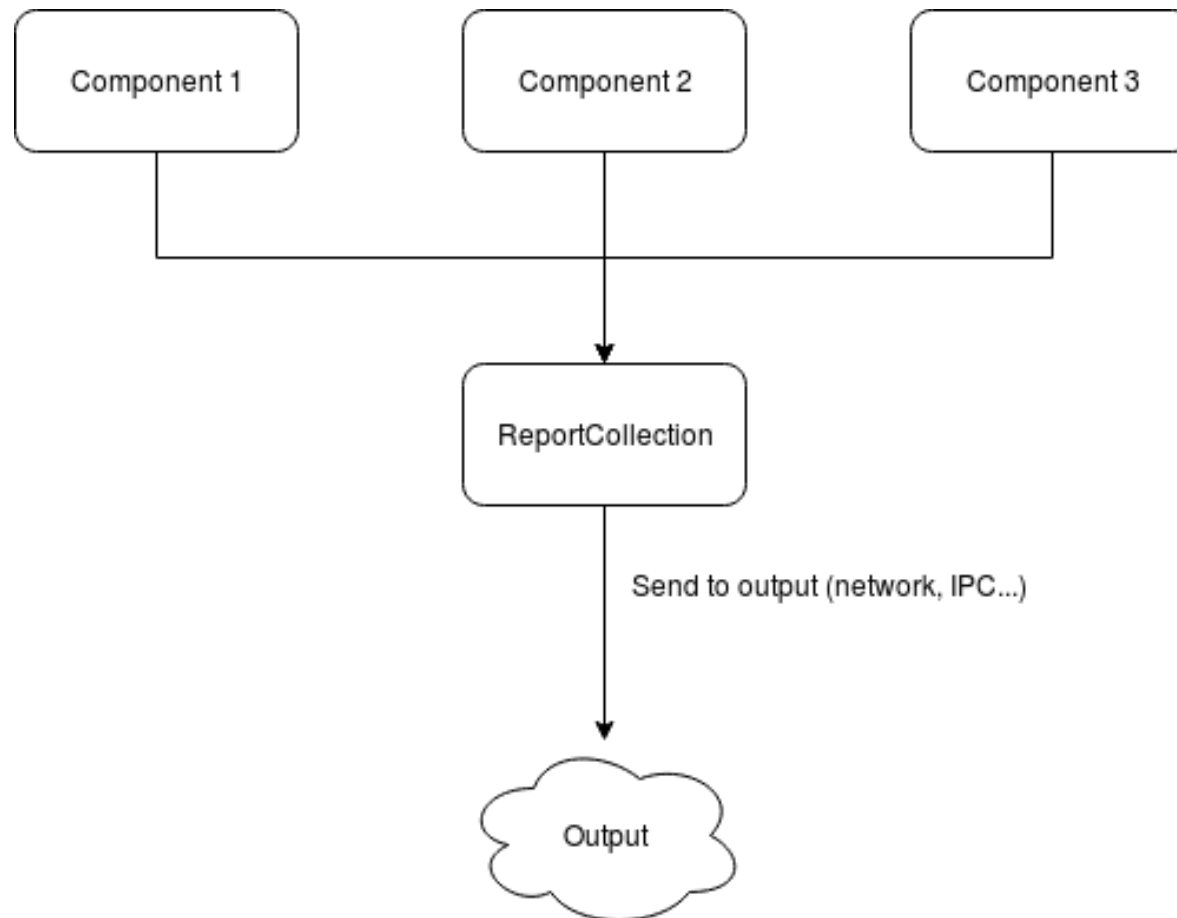
- Write:
  myChan ← x

Unbuffered Channels
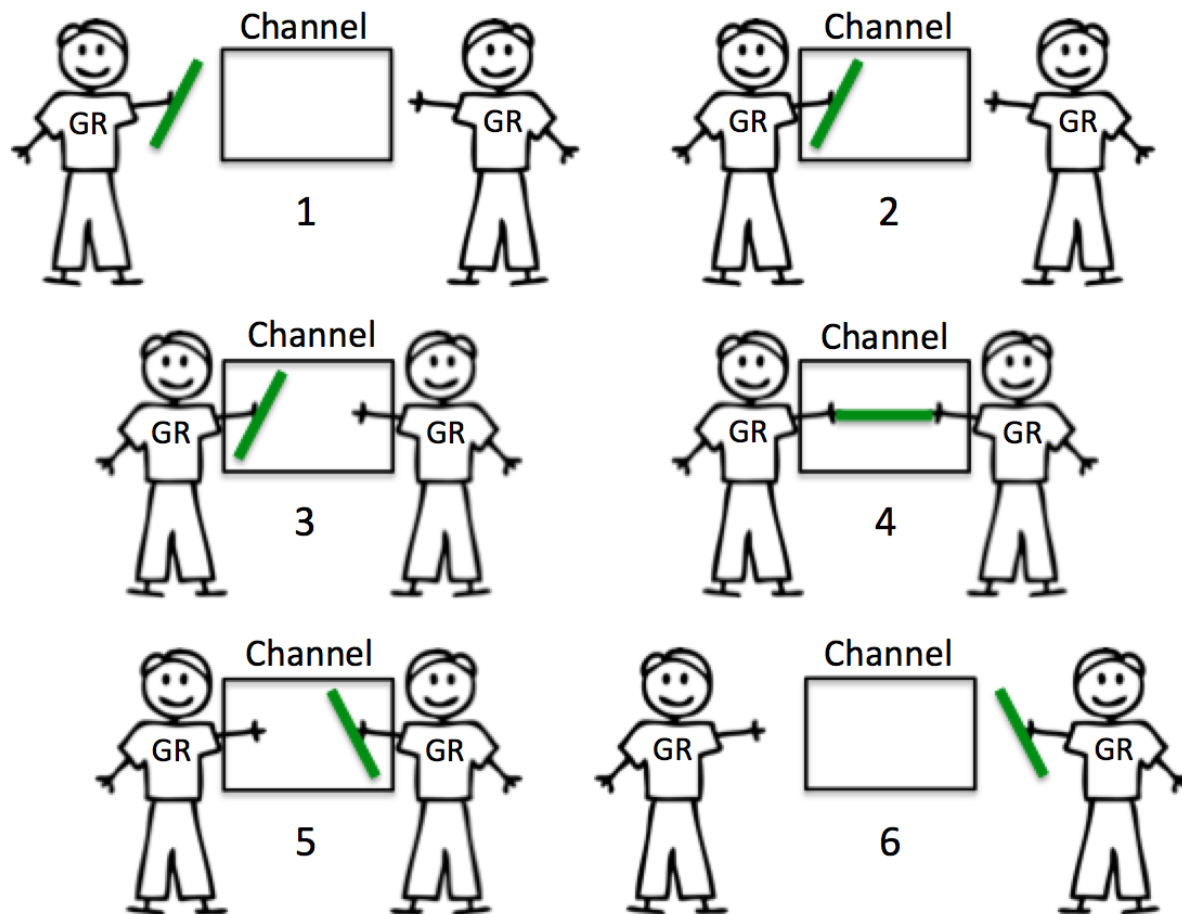
Buffered Channels

Example program

- Task: Write a program that collects reports of events from different soft- or hardware components. The components send the report in random intervals.

Architecture

Resources

- Official Website: https://golang.org/

- Language Design:
  https://talks.golang.org/2012/splash.article

- Detailed Explanation of Channels:
  https://www.ardanlabs.com/blog/2014/02/the-nature-of-channels-in-go.html

- Article about the Goroutine Scheduler:
  https://rakyll.org/scheduler/

Books

- Learning the language:
    - Introducing Go
    - The Go Programming Language (extensive)

- Recommendation if you want to get stuff done:
    - Go in Action

- The books are available on the shared server in the book section (buecher/go)

- The other books are domain specific or of questionable quality

Source of Images

- Unbuffered Channels: https://www.ardanlabs.com/blog/images/goinggo/Screen+Shot+2014-02-16+at+10.10.54+AM.png
- Buffered Channels: https://www.ardanlabs.com/blog/images/goinggo/Screen+Shot+2014-02-17+at+8.38.15+AM.png
-