

Discover Go's beauty of concurrency

SKG Gophers Meetup - October 2021

Patroklos Papapetrou (@softwaregarden)

Some ego boost

- Working from home the last 7 years
- Principal software engineer (last 5 mainly in golang)
- Clean code evangelist
- 25+ years in software engineering
- Father of two
- Organizer of various tech events in Greece

Today's topics

- Theoretical background
- Goroutines and wait groups
- Mutex flavours
- Channels
- Q&A

Some (essential) theoretical background

- Concurrency VS Parallelism
 - Some times blended / some times completely opposite
- Go scheduler
 - Orchestration
 - Synchronisation
- Hardware threads VS OS threads
- The trick of scheduling period

Go-routines

- Functions or Methods
- Run concurrently / in parallel with others
- Extremely light-weight (especially compared to threads)
- Managed by go runtime package
- Run in the same address space (thus synchronisation is required)

CODING TIME

Mutex

- Orchestration / locking mechanism
- Eliminated data races
- They come in two flavours
 - Standard (lock everything)
 - Read-Write (allow reads, lock writes)
- There's no free beer!

CODING TIME

Channels

- What a channel is not?
 - Data structure
 - Queue
- What a channel is?
 - Primitive language type
 - Allows signalling between go-routines
- Flavours
 - Buffered vs Unbuffered
- States
 - Open, nil or zero value and closed

CODING TIME

THANK YOU - QUESTIONS?