

Concurrencia y Go rutinas

Workshop de Go

7 de abril 2016

Iván Jaimes

Backend developer, Iver

Workshop de Go

1.1. Concurrencia ¿Qué es?

Es la composición de los cálculos de ejecución de forma independiente.

Es una manera de estructurar el software, en particular como una forma de describir

código limpio que interactúa bien con el mundo real.
No es paralelismo.

Workshop de Go

2.1. Concurrencia no es paralelismo

Concurrencia no es el paralelismo, aunque permite el paralelismo.

Si se tiene sólo un procesador, el programa todavía puede ser concurrentes, pero no

puede ser paralelas.

Por otro lado, un programa concurrente bien escrito puede funcionar de manera eficiente en paralelo en un multiprocesador.

Para más información sobre esta distinción, véase el siguiente enlace.

golang.org/s/concurrency-is-not-parallelism (golang.org/s/concurrency-is-not-parallelism)

Workshop de Go

3.1. Concurrencia comunmente

- Un solo código, multiples instancias ejecutandose
- Un hilo por instancia (generalmente)

- Desarrollador describe explícitamente que pasa y cuando

Workshop de Go

4.1. Concurrencia en go

- Un solo código
- Una instancia en ejecución (si, solo una)

- Memoria compartida
- Desarrolladores describen que pasa al mismo tiempo
- Go decide en tiempo de ejecución como pasa
- Concurrencia no es paralellismo

Workshop de Go

5.1. Concurrencia en go

Tres elementos principales

- Goroutines (go keyword) describe las unidades que lo ejecutan

independientemente.

- Channels (chan type) comunicación entre goroutines
- Select elige entre los channels

El paquete sync provee las primitivas de sincronización (sync.Mutex, sync.WaitGroup)

Workshop de Go

6.1. Conclusiones

Las goroutines y los channels hacen más fácil el expresar operaciones complejas.

- Múltiples entradas

- Múltiples salidas
- Timeouts
- Fallas

Y son divertidas!

Thank you

7 de abril 2016

Tags: Go (#ZgotmplZ)

Iván Jaimes

Backend developer, Iver

