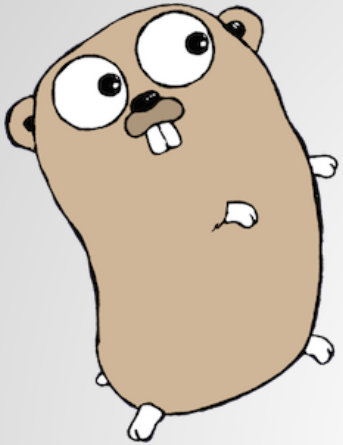
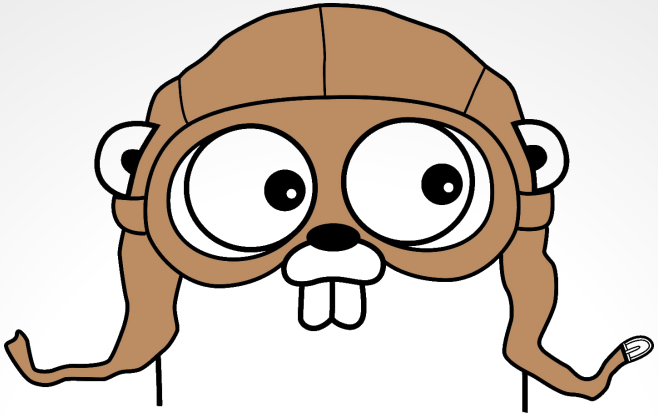


Concurrency With Go!

Come for the gophers, stay for the concurrency

John-Alan Simmons



John-Alan Simmons

CTO @ ConferenceCloud

 **@jsimnz**

 **@iamjsimnz**

Introduction

```
if new_to_go || played_with_go {
   fmt.Println(“Keep watching!”)
} else {
   fmt.Println(“Watch with at least 1 eye!”)
}
```

What is concurrency?

- Concurrency != Parallelism
- A way to structure and compose independently executing pieces of code.

Why the hell do we need it?

“ Look around you. What do you see?

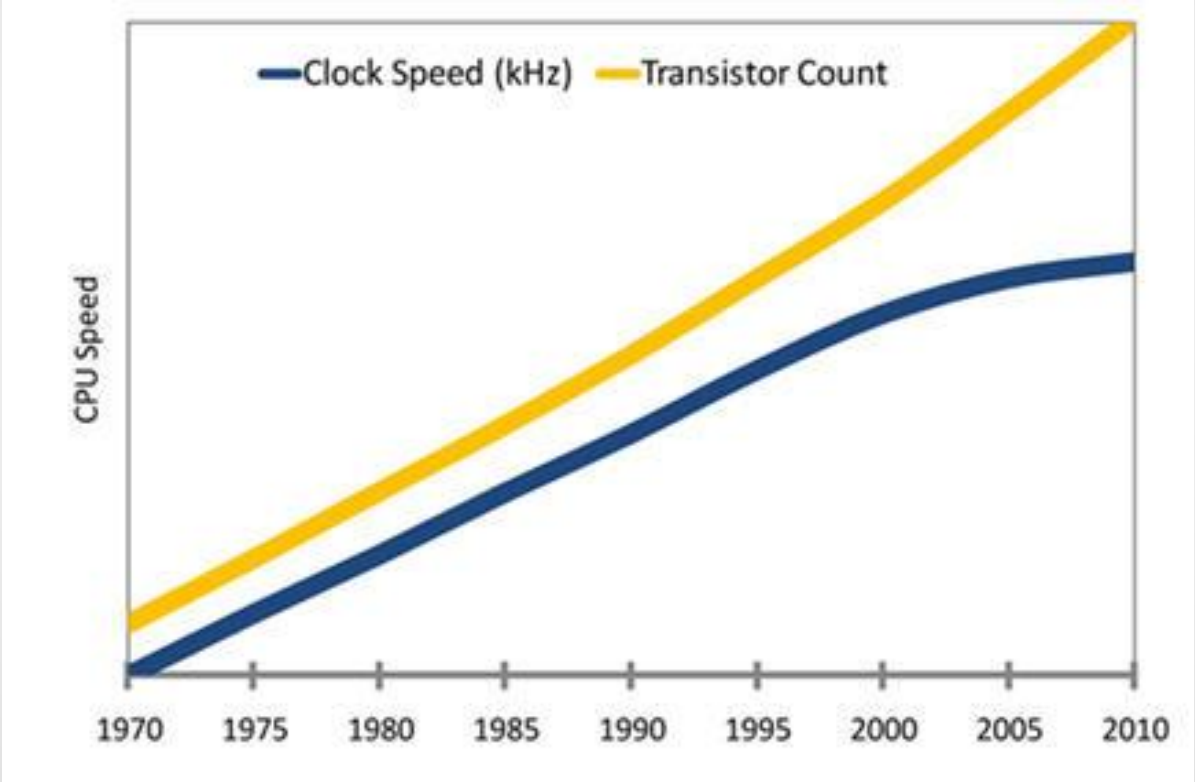
Do you see a single-stepping world doing one thing at a time?

Or do you see a complex world of interacting, independently behaving pieces?

Sequential processing on its own does not model the world's behavior. ”

- Rob Pike, Co-creator of Go

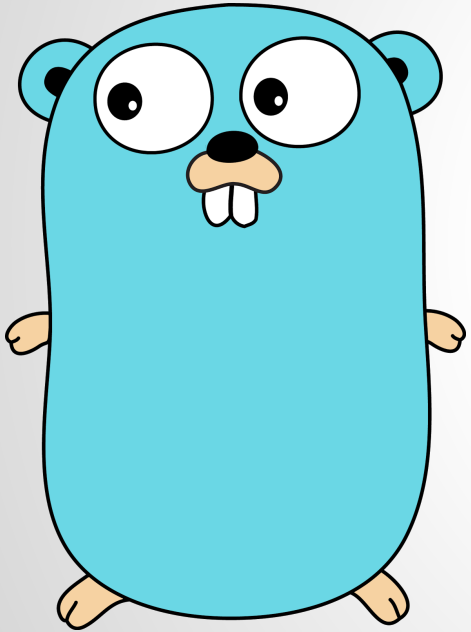
Moore's Law....DEAD



The old way of doing things

- OS Threads
- Multiple Processes
- Global Locks
- Shared Memory ←A nightmare

The new of doing things



Go(lang)!

Go(lang) one sheet

- 5 Years old. STABLE!
- Garbage Collected
- Small Language syntax
- Statically Compiled. Single Binary
- Strongly Typed
- Goroutines (Green Threads)
- Channels

Powerful Concurrency Primitives: Goroutines

- Green Threads
- Lighter than OS Threads
- 8K Memory
- Easy to execute

Powerful Concurrency Primitives: Channels

“ Don't communicate by sharing memory;
share memory by communicating. “

- Go Team

Powerful Concurrency Primitives: Channels

- Inter-goroutine communication
- Buffered/Non-Buffered
- Bidirectional
- Similar to OS Pipes

Let’s jump in

You're homework

- Google I/O 2013 - Advanced Concurrency Patterns
  - http://bit.ly/1DleBIv
- Google I/O 2012 - Go Concurrency Patterns
  - http://bit.ly/1EsGvG3
- Rob Pike - Concurrency Is Not Parallelism'
  - http://bit.ly/1DcWhAW

Golang gopher thanks you!

Questions?