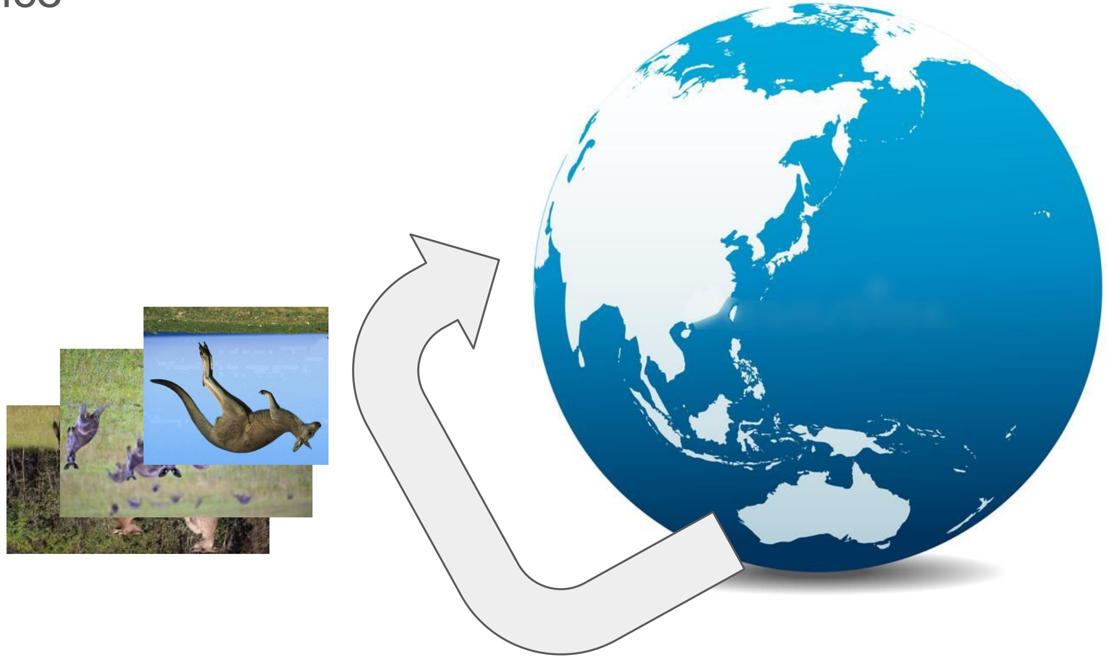
kangaroos

let’s flip pixels

Friend sends me photos

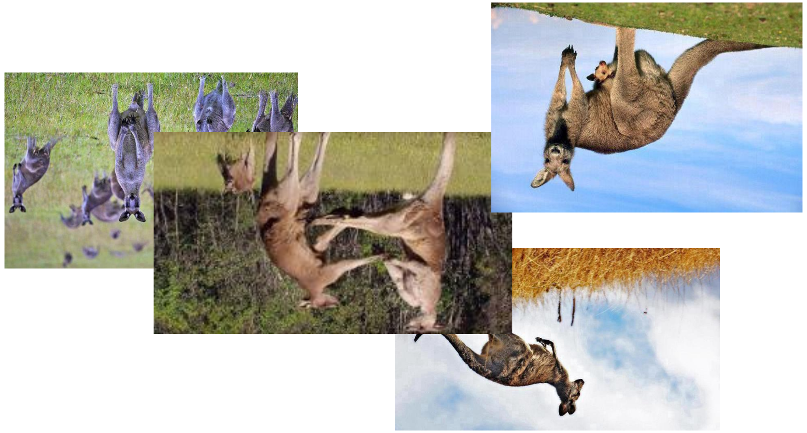
From Australia to France



But they look upside down

But they look upside down

Which makes sense,
because they were taken
in Australia

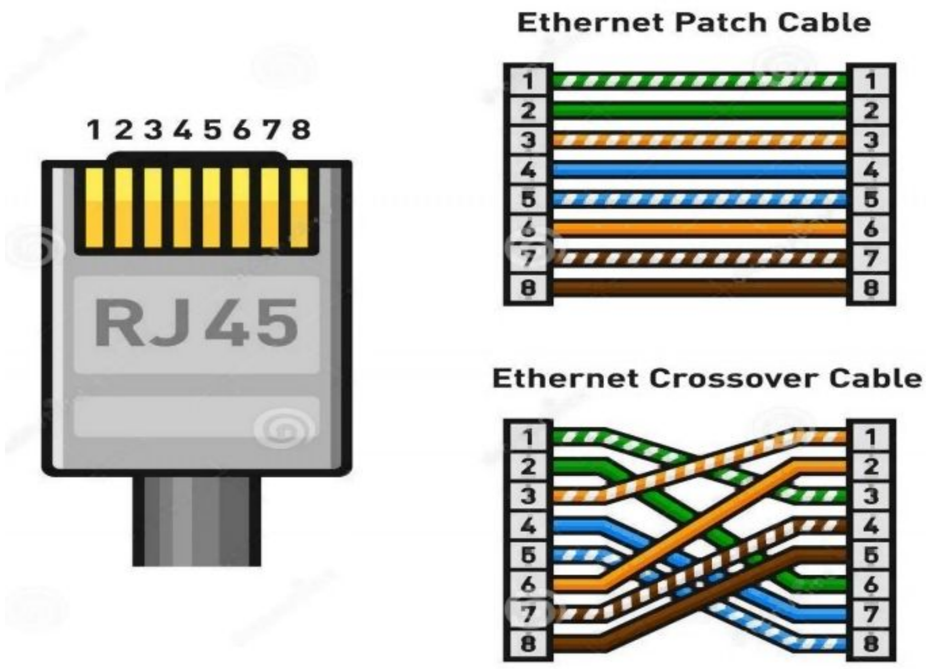


But they look upside down

Which makes sense, because they were taken in Australia

But the photographer was also in Australia...

Crossover cable



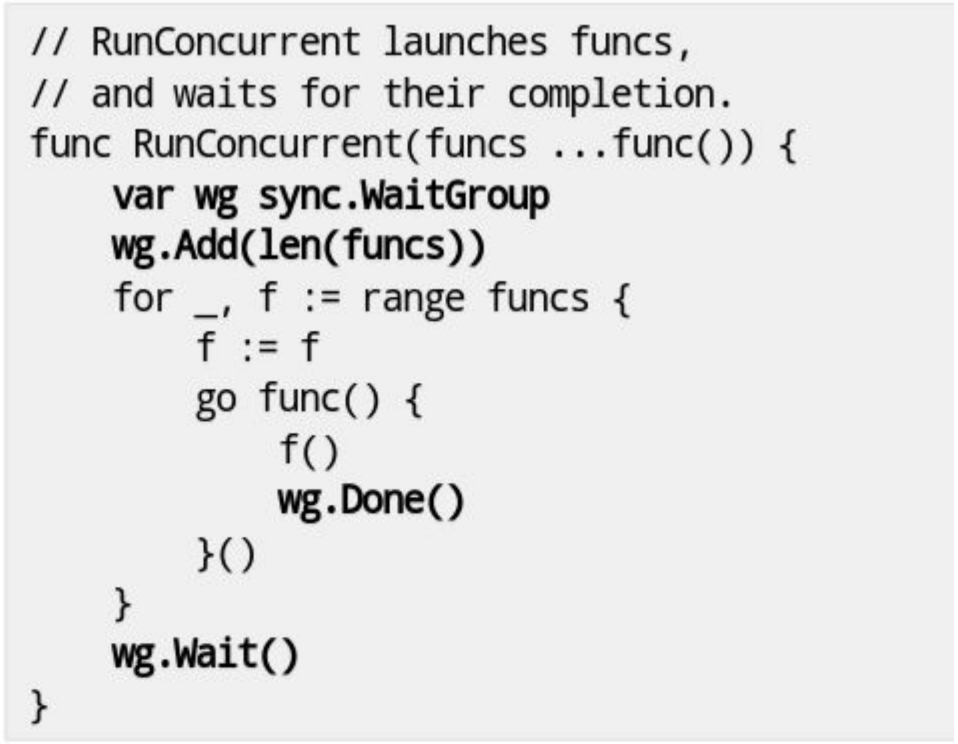
Fix

Let’s make a program

to flip the pictures back

(vertically)

sync.WaitGroup

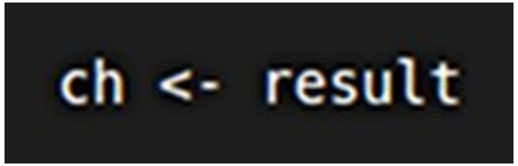


```
// RunConcurrent launches funcs,
// and waits for their completion.
func RunConcurrent(funcs ...func()) {
    var wg sync.WaitGroup
    wg.Add(len(funcs))
    for _, f := range funcs {
        f := f
        go func() {
            f()
            wg.Done()
        }()
    }
    wg.Wait()
}
```

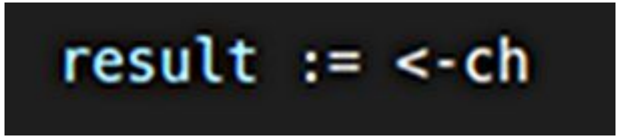
Channels

Producer → **to** → **Consumer**

(writer) **(reader)**



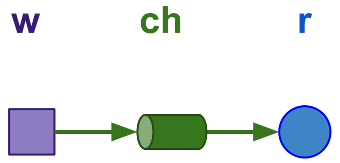
```
ch <- result
```



```
result := <-ch
```

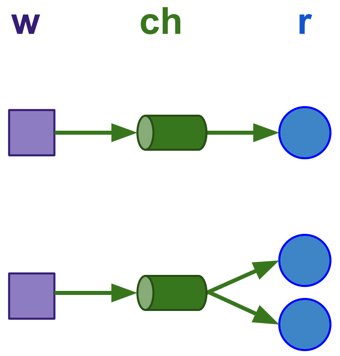
Channels

- 1 to 1 ✓easy
-
-
-



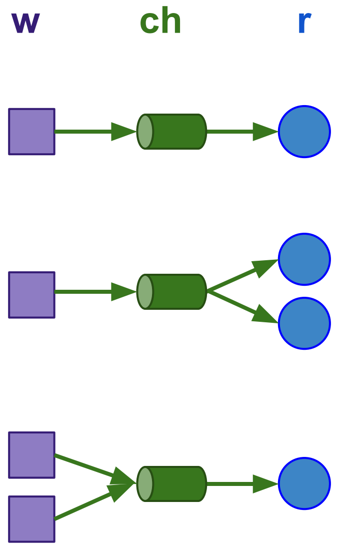
Channels

- 1 to 1 ✓easy
- 1 to **N** ✓easy
-
-



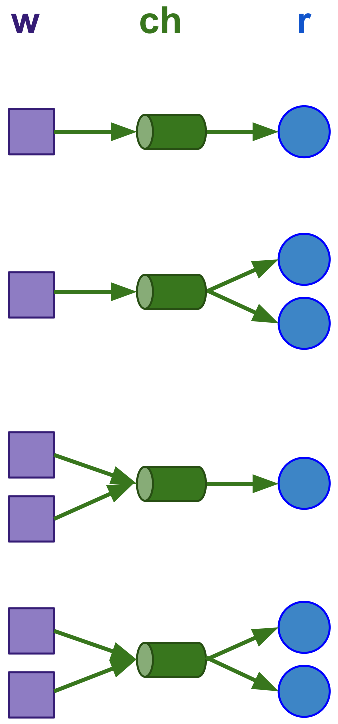
Channels

- 1 to 1 ✓easy
- 1 to **N** ✓easy
- **M** to 1 ⚠tricky
-



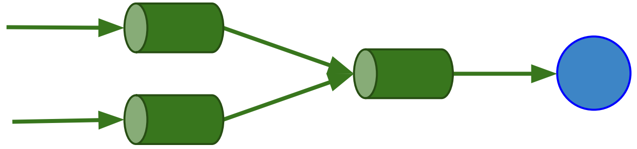
Channels

- 1 to 1 ✓easy
- 1 to **N** ✓easy
- **M** to 1 ⚠tricky
- **M** to **N** ⚠tricky

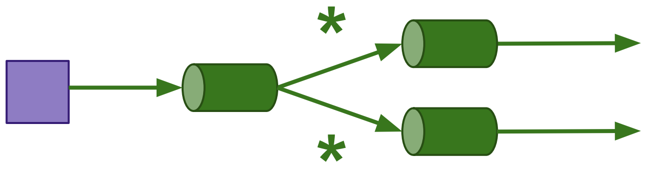


Channels

Fan-in



Fan-out



Easy to implement (but be cautious)

Exercise

```
$ go build -o exo

$ time ./exo
```

Exercise : benchmarks

```
$ go test -bench=.
```

Exercise : Pprof

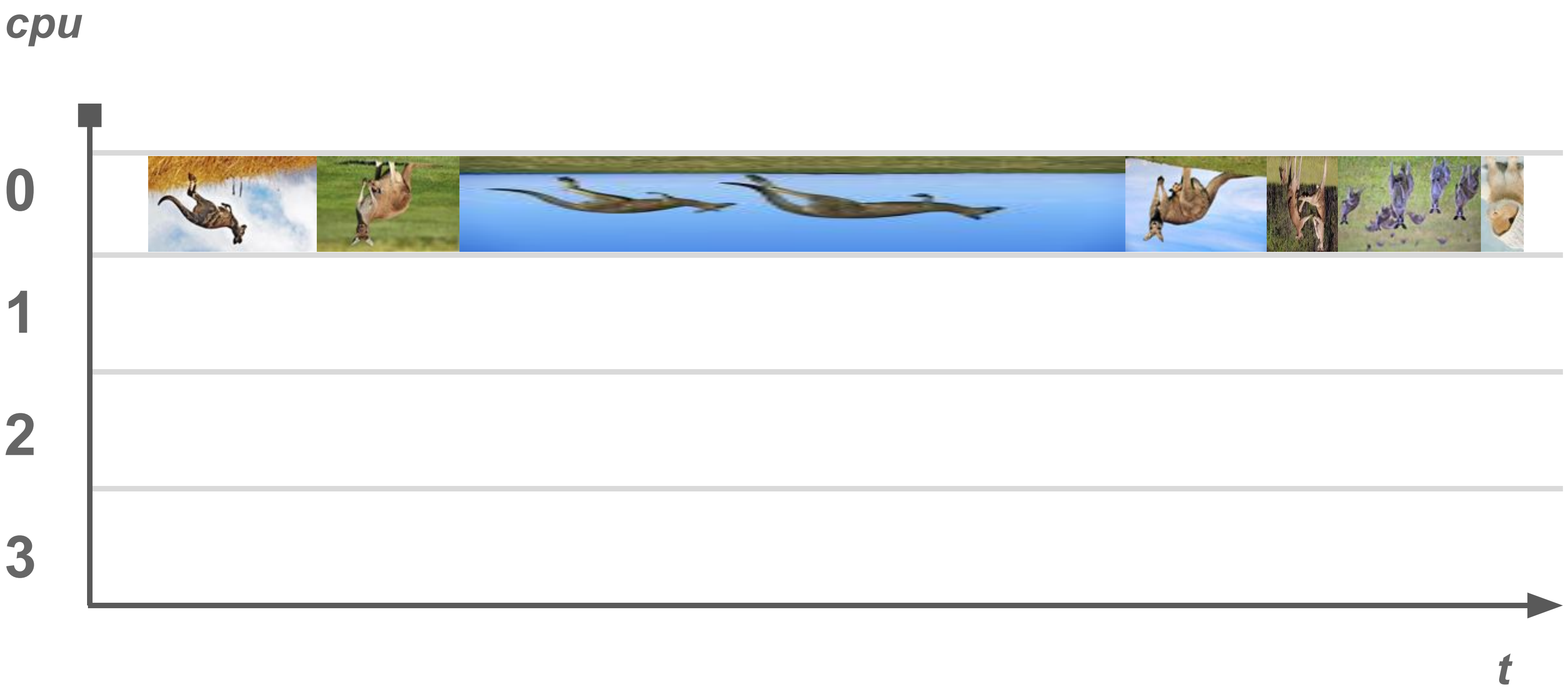
```
$ go test -bench=BenchmarkFlipA -cpuprofile A.prof

$ go tool pprof -svg A.prof > A.svg
```

Exercise : Trace

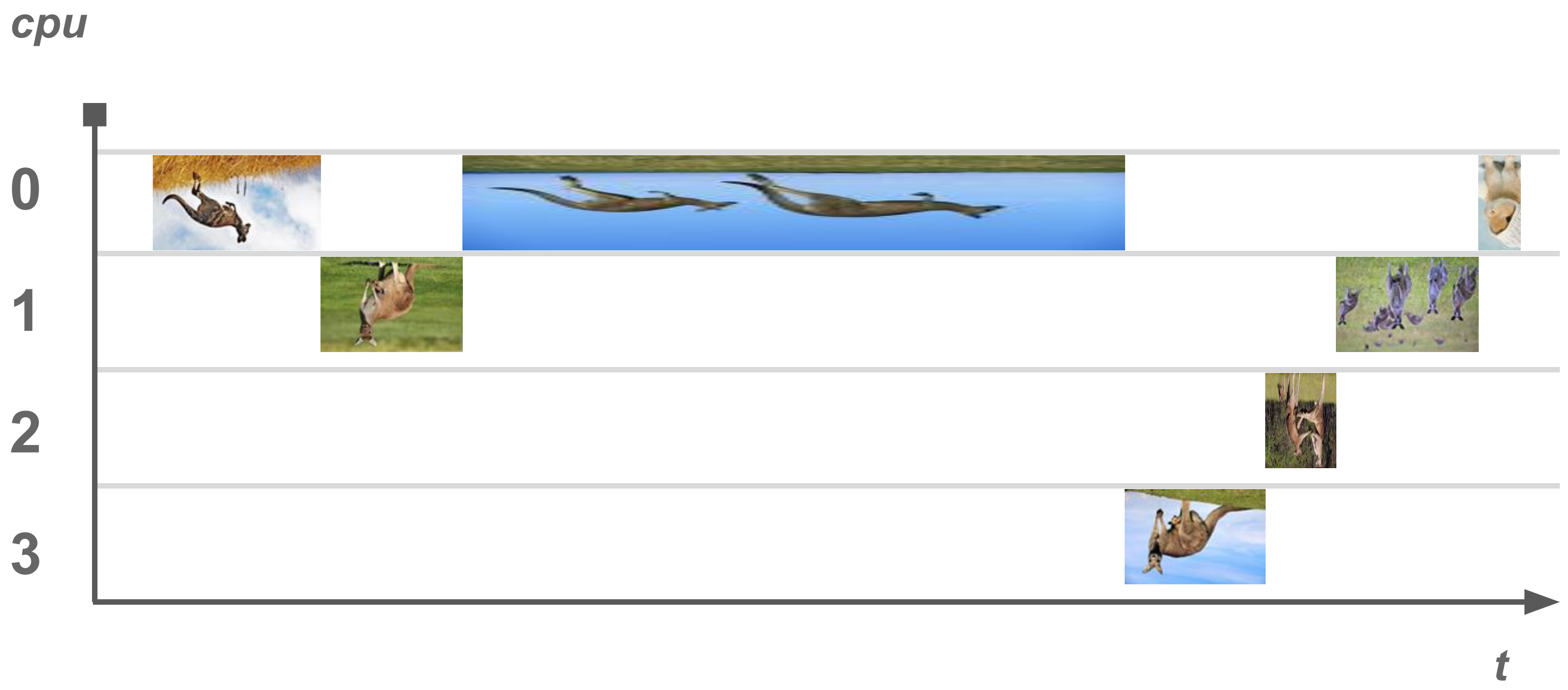
```
$ go test -bench=BenchmarkFlipA -trace A.out

$ go tool trace A.out
```

Works mostly in Chrome

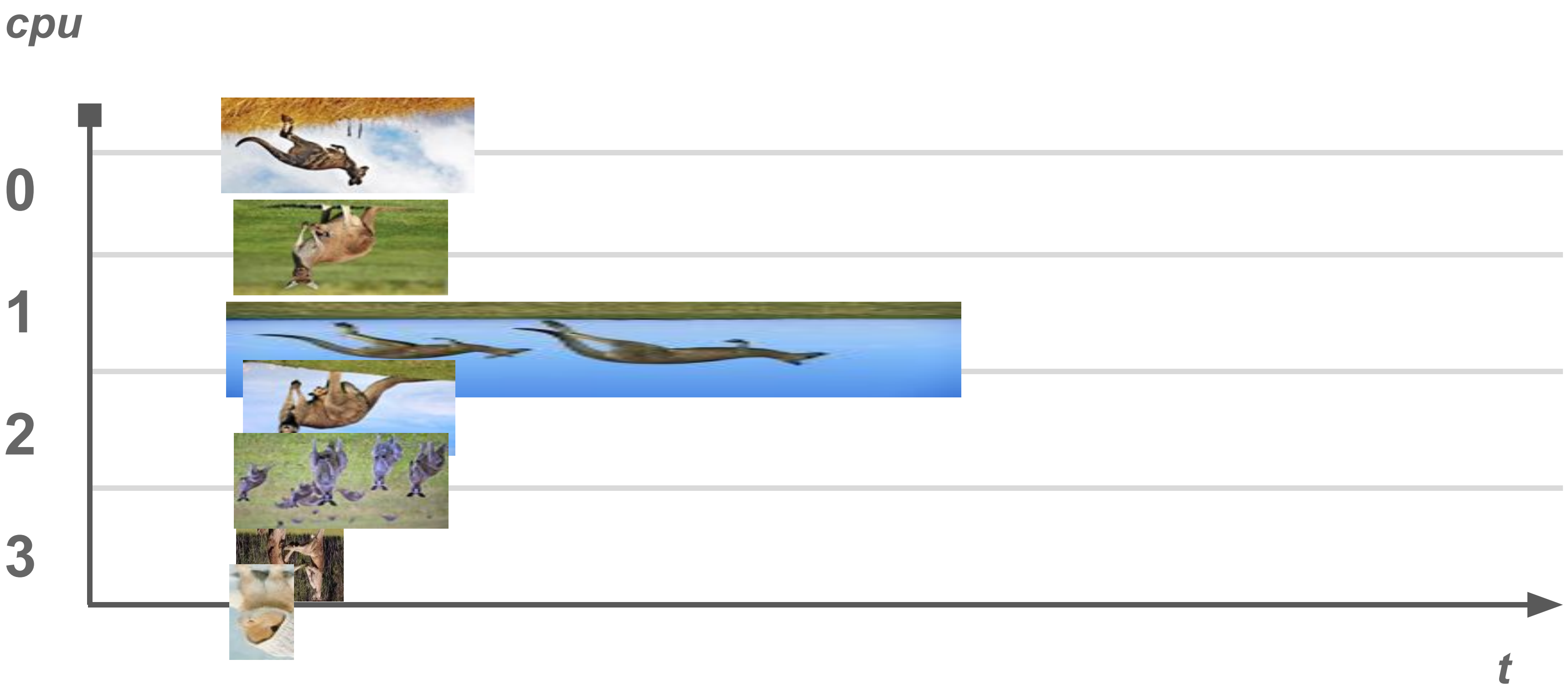
Flip : strategy A



Flip : strategy A

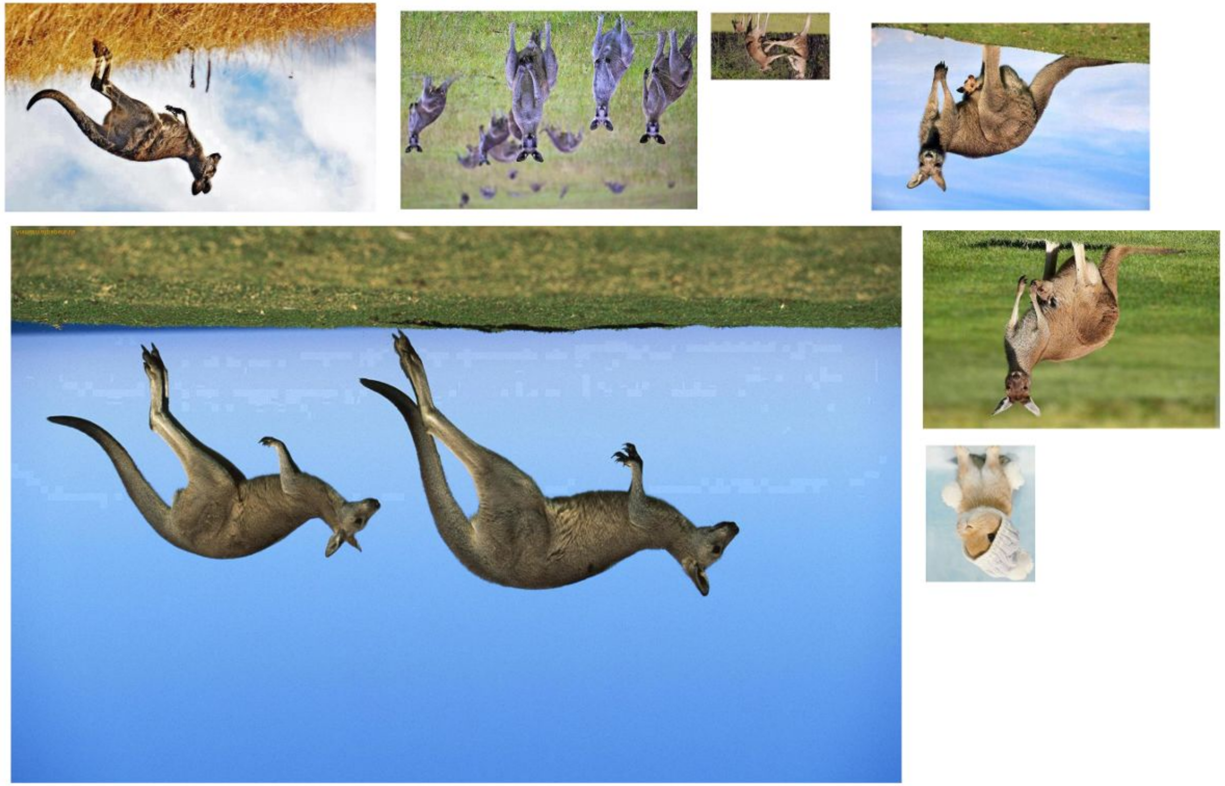


Flip : strategy B



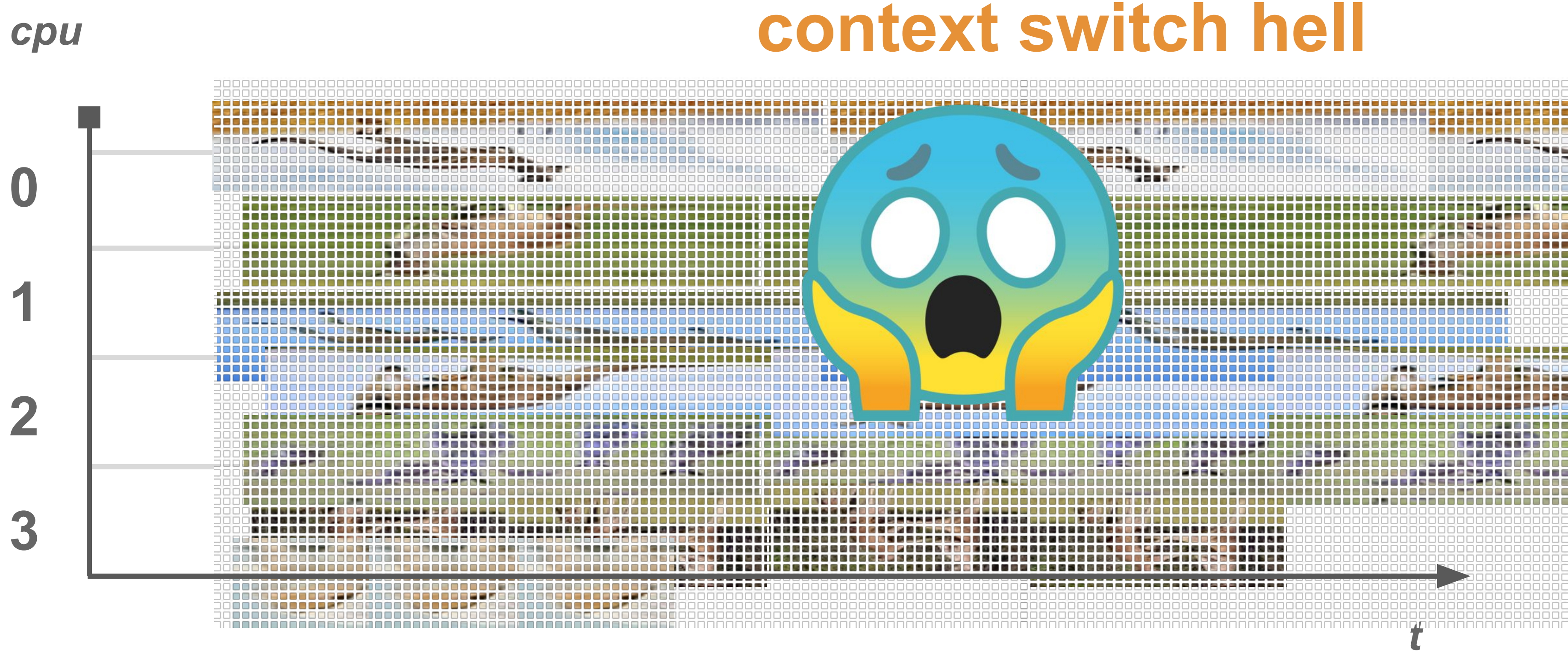
Tasks size

Not all photos have the same amount of pixels!



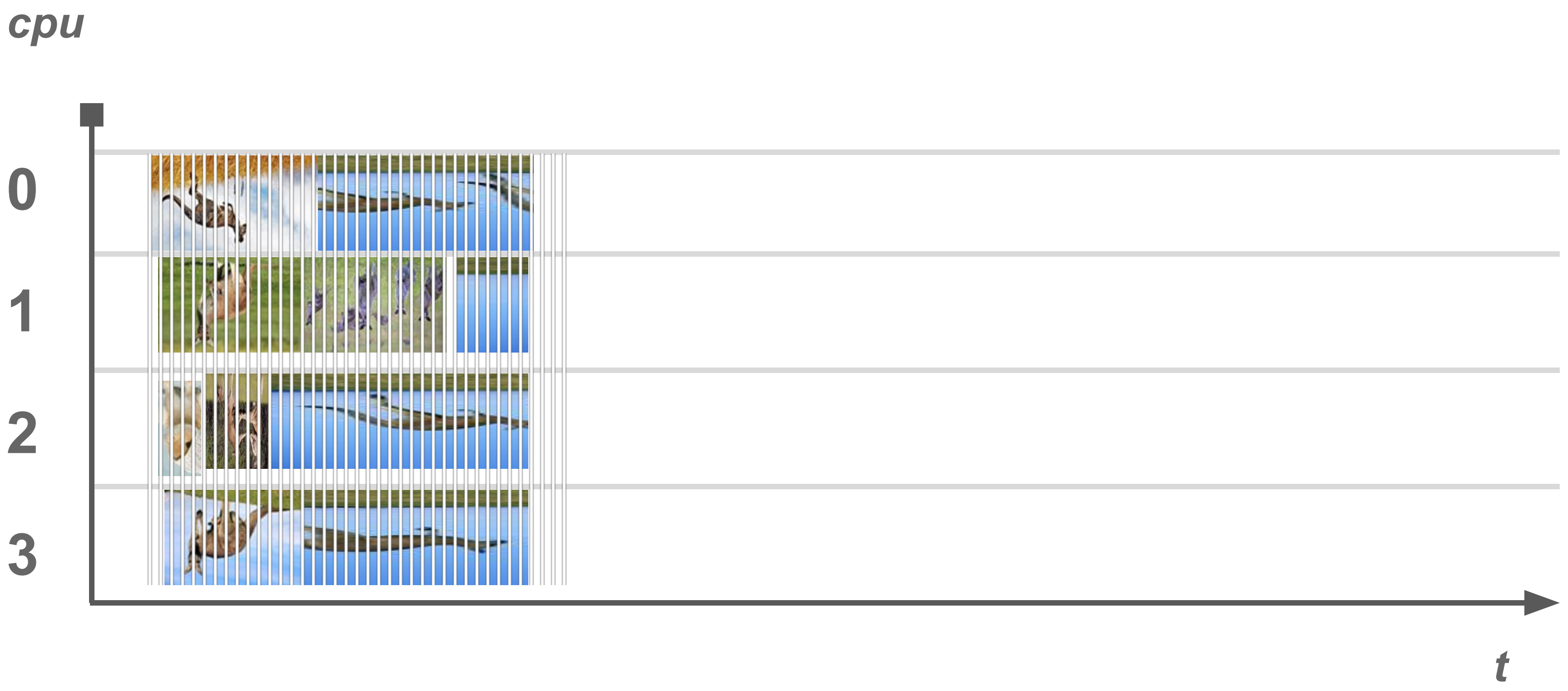
Flip : strategy C

1 goroutine/pixel ??
context switch hell



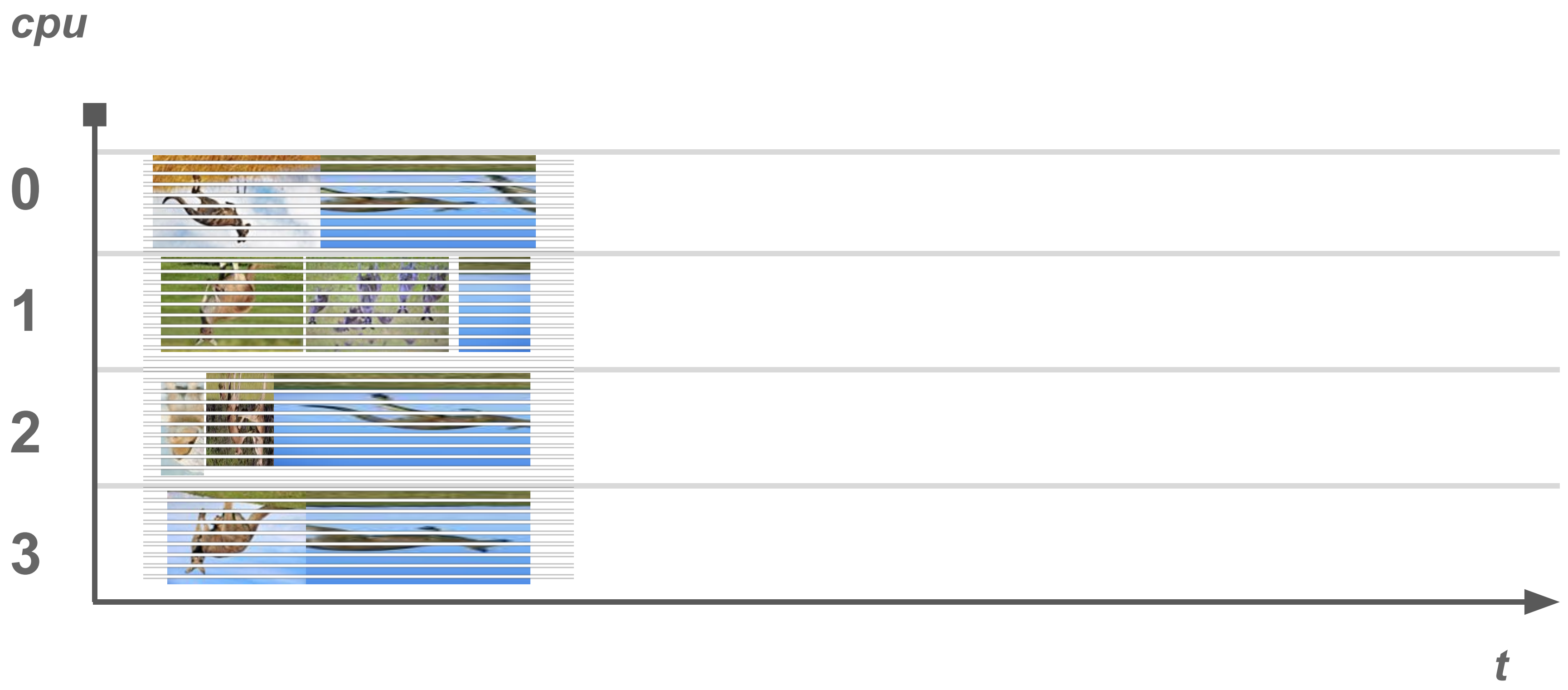
Flip : strategy D

1 goroutine/column



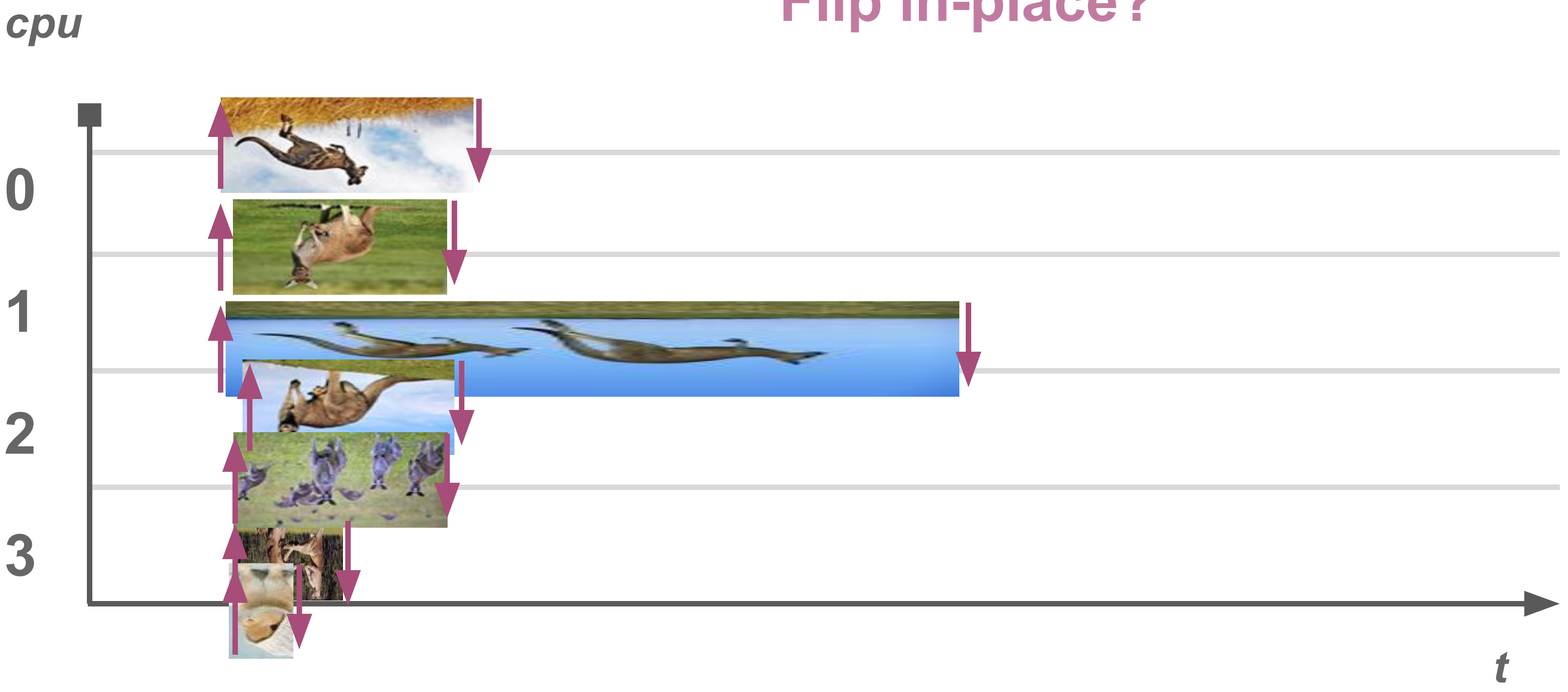
Flip : strategy E

1 goroutine/line

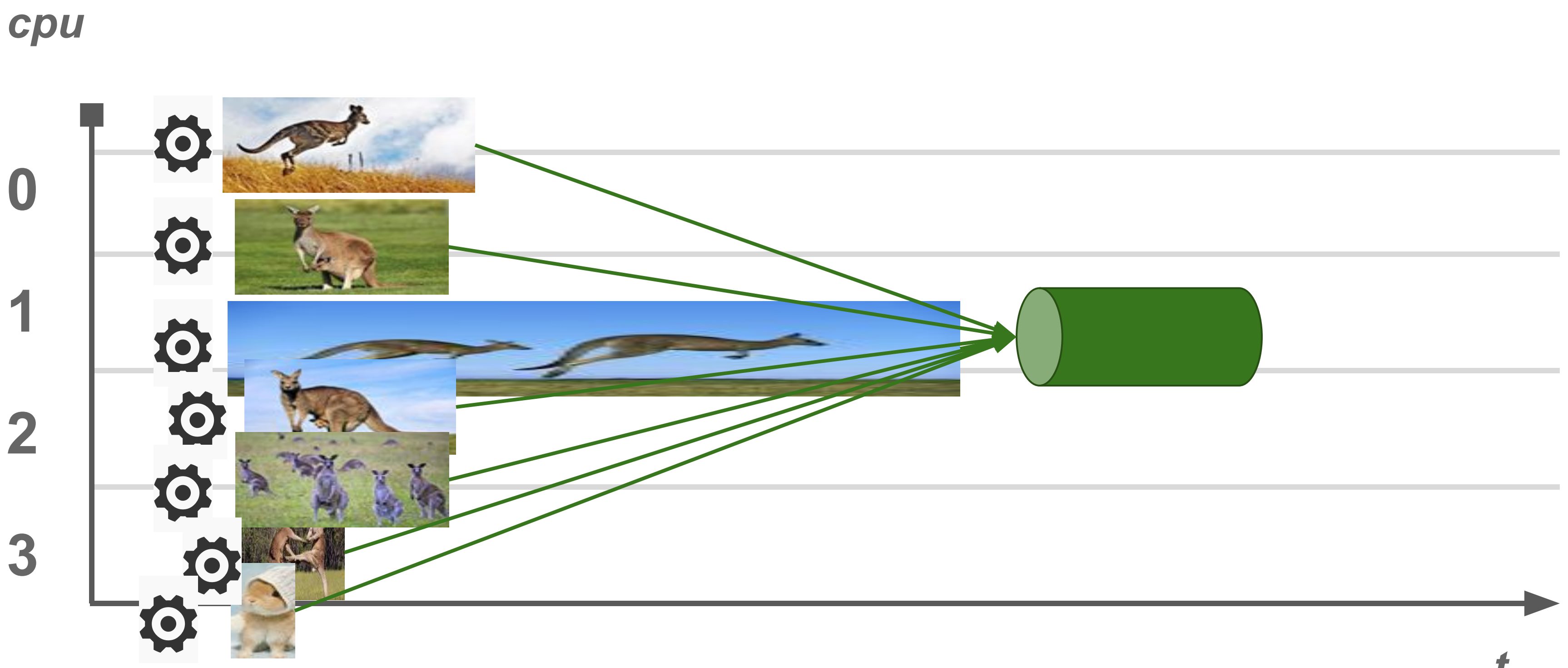


Flip : strategy F

Flip in-place?



Flip : strategy G



Takeaways

@val_deleplace

- `go tool pprof` is 👍
- `go tool trace` is 👍
- concurrency is hard to get right
- goroutines and channels are useful (but tricky)
- sync.WaitGroup is 👍