Table of Contents

Preface. vii

1. An Introduction to Concurrency. 1

Moore's Law, Web Scale, and the Mess We're In 2

Why Is Concurrency Hard? 4

Race Conditions 4

Atomicity 6

Memory Access Synchronization 8

Deadlocks, Livelocks, and Starvation 10

Determining Concurrency Safety 18

Simplicity in the Face of Complexity 20

2. Modeling Your Code: Communicating Sequential Processes. 23

The Difference Between Concurrency and Parallelism 23

What Is CSP? 26

How This Helps You 29

Go's Philosophy on Concurrency 31

3. Go's Concurrency Building Blocks. 37

Goroutines 37

The sync Package 47

WaitGroup 47

Mutex and RWMutex 49

Cond 52

Once 57

Pool 59

Channels 64

The select Statement 78

www.allitebooks.com

The GOMAXPROCS Lever 83
Conclusion 83

4. Concurrency Patterns in Go. 85
Confinement 85
The for-select Loop 89
Preventing Goroutine Leaks 90
The or-channel 94
Error Handling 97
Pipelines 100
Best Practices for Constructing Pipelines 104
Some Handy Generators 109
Fan-Out, Fan-In 114
The or-done-channel 119
The tee-channel 120
The bridge-channel 122
Queuing 124
The context Package 131
Summary 145

5. Concurrency at Scale. 147
Error Propagation 147
Timeouts and Cancellation 155
Heartbeats 161
Replicated Requests 172
Rate Limiting 174
Healing Unhealthy Goroutines 188
Summary 194

6. Goroutines and the Go Runtime. 197
Work Stealing 197
Stealing Tasks or Continuations? 204
Presenting All of This to the Developer 212
Conclusion 212

A. Appendix. 213

Index. 219

www.allitebooks.com