

УДК 004.432  
ББК 32.972.1  
Э98

**Титмус М. А.**

Э98 Облачный Go / пер. с англ. А. Н. Киселева. – М.: ДМК Пресс, 2022. – 418 с.: ил.

**ISBN 978-5-97060-965-1**

Go – первый язык программирования, спроектированный специально для разработки облачных приложений. В настоящее время он занял лидирующие позиции в облачной разработке и используется повсюду: от Docker до Harbour, от Kubernetes до Consul, от InfluxDB до CockroachDB.

Требования к масштабированию вынуждают разработчиков размещать свои сервисы на десятках и сотнях серверов – IT-отрасль постепенно становится «облачной». Но как разрабатывать и поддерживать такой сервис? В этой книге описывается практическая реализация сложных принципов проектирования облачных вычислений с помощью Go. Издание адресовано опытным разработчикам, особенно инженерам веб-приложений и инженерам по надежности, которые решают задачи управления и развертывания облачных приложений.

УДК 004.432  
ББК 32.972.1

Authorized Russian translation of the English edition of Cloud Native Go ISBN 9781492076339. This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same. Russian language edition copyright © 2022 by DMK Press. All rights reserved.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN 978-1-492-07633-9 (англ.)  
ISBN 978-5-97060-965-1 (рус.)

© Matthew A. Titmus, 2021  
© Перевод, оформление, издание,  
ДМК Пресс, 2022

# Содержание

<b>От издательства</b> .....	17
<b>Об авторе</b> .....	18
<b>Об иллюстрации на обложке</b> .....	19
<b>Предисловие</b> .....	20
 <b>Часть I. ОБЛАЧНОЕ ОКРУЖЕНИЕ</b> .....	24
<b>Глава 1. Что такое «облачное» приложение?</b> .....	25
История развития до настоящего времени.....	26
Что значит быть «облачным»? .....	28
Масштабируемость.....	28
Слабая связанность .....	29
Устойчивость .....	30
Управляемость .....	32
Наблюдаемость .....	33
Что особенного в облачном окружении?.....	34
Итоги.....	35
 <b>Глава 2. Почему Go правит облачным миром</b> .....	36
Как появился Go .....	36
Особенности облачного мира .....	37
Композиция и структурная типизация .....	37
Понятность .....	39
Модель взаимодействия последовательных процессов .....	40
Быстрая сборка.....	41
Стабильность языка.....	42
Безопасность памяти .....	42
Производительность .....	43
Статическая компоновка.....	44
Статическая типизация .....	45
Итоги.....	46
 <b>Часть II. ОБЛАЧНЫЕ КОНСТРУКЦИИ В GO</b> .....	47
<b>Глава 3. Основы языка Go</b> .....	48
Базовые типы данных .....	48
Логические значения .....	49

Простые числа .....	49
Комплексные числа .....	50
Строки .....	50
Переменные .....	51
Сокращенная форма объявления переменных .....	51
Нулевые значения.....	52
Пустой идентификатор .....	53
Константы .....	54
Контейнеры: массивы, срезы и ассоциативные массивы.....	54
Массивы.....	55
Срезы .....	55
Работа со срезами.....	56
Оператор извлечения среза.....	58
Строки и срезы.....	58
Ассоциативные массивы .....	59
Проверка наличия в ассоциативном массиве .....	60
Указатели.....	61
Управляющие структуры .....	62
Забавный цикл for.....	63
Универсальная инструкция for .....	63
Обход в цикле элементов массивов и срезов .....	64
Обход в цикле элементов ассоциативных массивов.....	65
Инструкция if.....	65
Инструкция switch .....	66
Обработка ошибок .....	67
Создание ошибки.....	68
Необычные особенности функций: переменное число параметров	
и замыкания .....	69
Функции .....	69
Несколько возвращаемых значений .....	69
Рекурсия.....	70
Отложенные вычисления .....	70
Указатели как параметры .....	72
Функции с переменным числом аргументов .....	73
Передача срезов в параметре с переменным числом значений.....	74
Анонимные функции и замыкания .....	74
Структуры, методы и интерфейсы .....	75
Структуры .....	76
Методы .....	77
Интерфейсы.....	78
Проверка типа.....	79
Пустой интерфейс .....	79
Композиция путем встраивания типов.....	80
Встраивание интерфейсов .....	80
Встраивание структур.....	81
Продвижение.....	81
Прямой доступ к встроенным полям .....	81
Самое интересное: конкуренция.....	82

Сопрограммы .....	82
Каналы .....	83
Блокировка канала .....	83
Буферизация каналов .....	84
Заккрытие каналов .....	84
Прием значений из канала в цикле .....	85
select .....	85
Реализация тайм-аутов для каналов .....	86
Итоги .....	87

## **Глава 4. Шаблоны программирования облачных приложений.....88**

Пакет context .....	89
Что может дать контекст .....	90
Создание контекста .....	91
Определение крайних сроков и тайм-аутов контекста .....	91
Определение значений в контексте запроса .....	92
Использование контекста .....	92
Структура этой главы .....	93
Шаблоны стабильности .....	94
Circuit Breaker (Размыкатель цепи) .....	94
Применимость .....	94
Реализация .....	95
Пример кода .....	96
Debounce (Антидребезг) .....	97
Применимость .....	97
Компоненты .....	98
Реализация .....	98
Пример кода .....	99
Retry (Повтор) .....	101
Применимость .....	101
Компоненты .....	102
Реализация .....	102
Пример кода .....	102
Throttle (Дроссельная заслонка) .....	104
Применимость .....	104
Компоненты .....	105
Реализация .....	105
Пример кода .....	106
Timeout (Тайм-аут) .....	107
Применимость .....	107
Компоненты .....	107
Реализация .....	108
Пример кода .....	108
Шаблоны конкуренции .....	110
Fan-In (Мультиплексор) .....	110
Применимость .....	110
Компоненты .....	110
Реализация .....	110

Пример кода .....	111
Fan-Out (Демультимплексор) .....	112
Применимость .....	112
Компоненты .....	112
Реализация .....	113
Пример кода .....	113
Future (В будущем).....	114
Применимость .....	115
Компоненты .....	116
Реализация .....	116
Пример кода .....	116
Sharding (Сегментирование) .....	118
Применимость .....	118
Компоненты .....	119
Реализация .....	119
Пример кода .....	121
Итоги.....	124
<b>Глава 5. Конструирование облачной службы .....</b>	<b>125</b>
Давайте создадим службу!.....	125
Что такое хранилище пар ключ/значение? .....	126
Требования.....	126
Что такое идемпотентность, и почему это важно?.....	126
Конечная цель .....	128
Итерация 0: базовая функциональность.....	128
Наш суперпростой API .....	129
Итерация 1: монолит.....	130
Создание HTTP-сервера с использованием net/http.....	131
Создание HTTP-сервера с использованием gorilla/mux .....	132
Создание минимальной службы .....	133
Инициализация проекта с помощью модулей Go .....	133
Переменные в путях URI .....	134
Множество сопоставлений.....	135
Создание службы RESTful .....	135
Методы RESTful.....	136
Реализация функции создания .....	136
Реализация функции чтения .....	138
Добавление в структуру данных поддержки использования в конкурентном окружении .....	140
Интеграция мьютекса чтения/записи в приложение .....	141
Итерация 2: долговременное хранение ресурса .....	142
Что такое журнал транзакций? .....	143
Формат журнала транзакций.....	143
Интерфейс регистратора транзакций .....	144
Сохранение состояния в журнале транзакций.....	144
Создание прототипа регистратора транзакций .....	145
Определение типа события.....	146

Реализация FileTransactionLogger .....	148
Создание экземпляра FileTransactionLogger .....	149
Добавление записей в конец журнала транзакций .....	150
Использование bufio.Scanner для воспроизведения транзакций из журнала .....	151
Интерфейс регистратора транзакций (еще раз) .....	153
Инициализация FileTransactionLogger в веб-службе .....	153
Интеграция FileTransactionLogger в веб-службу .....	155
Будущие улучшения .....	155
Сохранение состояния во внешней базе данных .....	155
Работа с базами данных в Go .....	156
Импортирование драйвера базы данных .....	157
Реализация PostgresTransactionLogger .....	157
Создание экземпляра PostgresTransactionLogger .....	158
Выполнение SQL-запроса INSERT с помощью db.Exec .....	160
Использование db.Query для воспроизведения транзакций из журнала .....	161
Инициализация PostgresTransactionLogger в веб-службе .....	162
Будущие улучшения .....	163
Итерация 3: реализация безопасности транспортного уровня .....	163
Transport Layer Security .....	164
Сертификаты, центры сертификации и доверие .....	164
Закрытый ключ и файлы сертификатов .....	165
Формат Privacy Enhanced Mail (PEM) .....	165
Защита веб-службы с помощью HTTPS .....	166
В заключение о транспортном уровне .....	167
Контейнеризация хранилища пар ключ/значение .....	168
Основы Docker .....	169
Dockerfile .....	169
Сборка образа контейнера .....	170
Запуск образа контейнера .....	171
Проверка запущенного образа контейнера .....	172
Отправка запроса в опубликованный порт контейнера .....	173
Запуск нескольких контейнеров .....	174
Остановка и удаление контейнеров .....	174
Сборка контейнера для службы хранилища пар ключ/значение .....	175
Итерация 1: добавление двоичного файла в пустой образ .....	176
Итерация 2: многоэтапная сборка .....	178
Сохранение данных контейнера вовне .....	179
Итоги .....	180
<b>Часть III. ОБЛАЧНЫЕ АТТРИБУТЫ .....</b>	<b>182</b>
<b>Глава 6. Все дело в надежности .....</b>	<b>183</b>
В чем суть облачных вычислений? .....	184
Все дело в надежности .....	184

Что такое надежность, и почему она так важна? .....	185
Надежность обеспечивается не только операторами .....	187
Достижение надежности .....	188
Предотвращение неисправностей .....	190
Рекомендуемые практики программирования .....	190
Особенности языка .....	190
Масштабируемость .....	191
Слабая связанность .....	191
Отказоустойчивость .....	192
Устранение неисправностей .....	192
Проверка и тестирование .....	193
Управляемость .....	194
Прогнозирование неисправностей .....	194
Непреодолимая актуальность методологии «Двенадцать факторов» .....	194
I. Кодовая база .....	195
II. Зависимости .....	196
III. Конфигурация .....	196
IV. Сторонние службы .....	198
V. Сборка, выпуск, выполнение .....	199
VI. Процессы .....	200
VII. Изоляция данных .....	200
VIII. Масштабируемость .....	201
IX. Живучесть .....	202
X. Сходство окружений разработки/эксплуатации .....	202
XI. Журналирование .....	203
XII. Задачи администрирования .....	204
Итоги .....	205
<b>Глава 7. Масштабируемость</b> .....	206
Что такое масштабируемость? .....	207
Различные формы масштабирования .....	208
Четыре основных узких места .....	209
С состоянием и без состояния .....	211
Состояние приложения и состояние ресурса .....	211
Преимущества отсутствия состояния .....	212
Отложенное масштабирование: эффективность .....	213
Эффективное кэширование с использованием кеша LRU .....	213
Эффективная синхронизация .....	217
Разделяйте память, общаясь .....	217
Уменьшение простоев на блокировках с помощью буферизованных каналов .....	219
Уменьшение простоев на блокировках с помощью сегментирования... ..	221
Утечки памяти могут вызвать... фатальную ошибку исчерпания памяти во время выполнения .....	222
Утечки сопрограмм .....	222
Вечно тикающие таймеры .....	223
В заключение об эффективности .....	225

Архитектуры служб.....	225
Архитектура монолитной системы.....	226
Архитектура системы микросервисов.....	227
Бессерверные архитектуры.....	229
Достоинства и недостатки бессерверных вычислений.....	229
Бессерверные службы.....	231
Итоги.....	233
<b>Глава 8. Слабая связанность.....</b>	<b>234</b>
Тесная связанность.....	235
Множество форм тесной связанности.....	236
Хрупкие протоколы обмена.....	236
Общие зависимости.....	237
Общий момент времени.....	237
Фиксированные адреса.....	238
Взаимодействия между службами.....	238
Шаблон обмена сообщениями запрос/ответ.....	239
Распространенные реализации шаблона запрос/ответ.....	240
Отправка HTTP-запросов с использованием net/http.....	240
Вызов удаленных процедур с использованием gRPC.....	244
Определение интерфейса с использованием протокола буферов.....	245
Установка компилятора протокола буферов.....	246
Определение структуры сообщения.....	247
Структура сообщений для взаимодействий с хранилищем пар ключ/значение.....	248
Определение методов службы.....	249
Компиляция протокола буферов.....	250
Реализация службы gRPC.....	251
Реализация клиента gRPC.....	253
Слабое связывание локальных ресурсов с помощью плагинов.....	255
Подключение плагинов с помощью пакета plugin.....	255
Словарь плагинов.....	256
Пример плагина.....	257
Интерфейс Sayer.....	257
Код плагина.....	258
Сборка плагинов.....	258
Использование плагинов Go.....	259
Запуск примера.....	261
Система плагинов HashiCorp для Go, доступных через RPC.....	261
Еще один пример плагина.....	262
Общий код.....	263
Реализация плагина.....	265
Процесс-потребитель.....	266
Гексагональная архитектура.....	269
Архитектура.....	269
Реализация гексагональной службы.....	270
Реорганизация компонентов.....	271



Наш первый разъем .....	272
Основное приложение.....	272
Адаптеры TransactionLogger .....	273
Порт FrontEnd.....	274
Все вместе .....	276
Итоги .....	277
<b>Глава 9. Устойчивость .....</b>	<b>279</b>
Почему устойчивость важна.....	280
Что подразумевается под сбоем системы? .....	281
Обеспечение устойчивости .....	282
Каскадные сбои .....	282
Предотвращение перегрузки .....	284
Дросселирование .....	284
Сброс нагрузки.....	288
Постепенное ухудшение качества обслуживания .....	289
Повтори еще раз: повторные запросы .....	289
Алгоритмы увеличения задержки.....	291
Размыкание цепи.....	294
Тайм-ауты.....	295
Использование контекста Context для реализации тайм-аутов на стороне службы.....	296
Прерывание ожидания обработки клиентских запросов HTTP/REST.....	298
Прерывание ожидания обработки клиентских запросов gRPC.....	299
Идемпотентность .....	301
Как сделать службу идемпотентной? .....	302
А как насчет скалярных операций? .....	303
Избыточность служб .....	304
Проектирование избыточности .....	305
Автоматическое масштабирование.....	307
Проверка работоспособности .....	308
Что подразумевается под «работоспособностью» экземпляра?.....	309
Три типа проверок работоспособности.....	309
Проверка жизнеспособности.....	310
Поверхностная проверка работоспособности .....	310
Глубокая проверка работоспособности.....	312
Открытие при отказе .....	313
Итоги .....	314
<b>Глава 10. Управляемость.....</b>	<b>315</b>
Что такое управляемость, и почему она важна? .....	316
Настройка приложения.....	317
Рекомендуемые приемы организации конфигураций.....	318
Настройка с использованием переменных окружения .....	319
Настройка с использованием аргументов командной строки .....	320
Стандартный пакет flag .....	320

Парсер командной строки Cobra.....	322
Настройка с использованием файлов.....	326
Наша структура конфигурационных данных.....	326
Формат JSON.....	327
Формат YAML .....	332
Наблюдение за изменениями в конфигурационных файлах .....	335
Viper: швейцарский армейский нож конфигурационных пакетов .....	340
Явно устанавливаемые значения в Viper.....	341
Работа с флагами командной строки в Viper .....	341
Работа с переменными окружения в Viper .....	342
Работа с конфигурационными файлами в Viper .....	342
Использование удаленных хранилищ пар ключ/значение в Viper.....	344
Значения по умолчанию в Viper.....	345
Управление функциональными возможностями с помощью флагов .....	345
Разработка флага для управления функциональной возможностью .....	346
Итерация 0: начальная реализация .....	347
Итерация 1: жестко запрограммированный флаг .....	347
Итерация 2: настраиваемый флаг .....	348
Итерация 3: динамический флаг .....	349
Динамические флаги как функции.....	350
Реализация функции динамического флага .....	350
Поиск функции флага .....	351
Функция маршрутизации .....	352
Итоги .....	353
<b>Глава 11. Наблюдаемость.....</b>	<b>354</b>
Что такое наблюдаемость?.....	355
Зачем нужна наблюдаемость? .....	355
Чем наблюдаемость отличается от «традиционного» мониторинга? .....	356
«Три столпа наблюдаемости».....	357
OpenTelemetry.....	358
Компоненты OpenTelemetry.....	359
Трассировка .....	360
Концепции трассировки .....	361
Трассировка с использованием OpenTelemetry .....	362
Создание экспортеров трассировки .....	364
Создание провайдера трассировки .....	366
Настройка глобального провайдера трассировки.....	367
Получение экземпляра трассировщика .....	367
Начальная и конечная операции .....	367
Установка метаданных операции .....	369
Автоматическое инструментирование .....	370
Собираем все вместе: трассировка .....	373
API-службы вычисления чисел Фибоначчи.....	374
Функция-обработчик службы вычисления чисел Фибоначчи.....	375
Функция main службы.....	376
Запуск служб.....	377

Вывод консольного экспортера .....	377
Просмотр результатов в Jaeger .....	378
Метрики.....	379
Два способа передачи метрик: принудительная и по запросу.....	381
Принудительная отправка метрик .....	382
Передача метрик по запросу .....	382
Какой подход лучше? .....	383
Метрики в OpenTelemetry.....	384
Создание экспортеров метрик .....	385
Установка глобального провайдера метрик.....	386
Экспортирование конечной точки метрик.....	386
Получение экземпляра Meter .....	388
Инструменты метрик.....	388
Собираем все вместе: метрики.....	394
Запуск служб.....	394
Вывод конечной точки метрик.....	395
Просмотр результатов в Prometheus .....	396
Журналирование .....	397
Рекомендуемые методы журналирования .....	397
Интерпретируйте журналы как потоки событий .....	398
Структурируйте события для последующего анализа .....	398
Лучше меньше, да лучше.....	400
Динамически фильтруйте журналируемые данные .....	400
Журналирование с использованием стандартного пакета log.....	401
Специальные функции журналирования .....	402
Журналирование в нестандартный объект записи .....	402
Флаги журналирования .....	403
Пакет журналирования Zap.....	403
Создание регистратора Zap .....	405
Журналирование с использованием Zap .....	405
Динамическая фильтрация журналируемых данных в Zap.....	407
Итоги .....	409
<b>Предметный указатель.....</b>	<b>410</b>