

*Посвящается моей семье Кэт и Хейли
за их терпение в течение всего времени,
пока я писал эту книгу.*

*Также посвящается Логану и Пенни,
моим постоянным коллегам по кодированию нечеткой логики,
которые помогли мне пройти через многие препятствия.
Счастливые воспоминания.*

Оглавление

Предисловие от издательства	14
Вступительное слово	15
Об авторе	16
О рецензентах	17
Предисловие	18
РАЗДЕЛ 1. ОБЩИЙ ОБЗОР APACHE ARROW, ЕГО ВОЗМОЖНОСТЕЙ, ПРЕИМУЩЕСТВ И ЦЕЛЕЙ	25
Глава 1. Начинаем работу с Apache Arrow	27
Технические требования	28
Изучение формата и спецификаций Arrow	28
Почему Arrow использует формат с представлением данных в памяти по столбцам	33
Изучение терминологии и схемы распределения физической памяти	36
Краткий обзор схем размещения в физической памяти, или TL;DR	38
Массивы фиксированной длины со значениями простейшего типа	39
Двоичные массивы с переменной длиной	40
Массивы двоичного представления с переменной длиной	41
Массивы типа «список» и «список фиксированного размера»	44
Массивы ListView	46
Массивы типа «структура»	47
Массивы типа Union – разреженные и плотные	48
Массивы, кодируемые как словари	50
Массивы с методом кодирования Run-End	51
Null-массивы	54
Как объясняться с Arrow	54
Управление версиями и стабильность формата Arrow	56
Нужно ли загружать библиотеку? Конечно, нужно	57
Настройка сферы деятельности	58
Использование pyarrow для Python	58
Изучение основ использования pyarrow	59
Создание массива структур	61
Использование пакетов записей и операций с нулевым копированием	63

Обработка отсутствующих значений.....	65
Упражнение для самостоятельного выполнения.....	65
C++ для 1337-кодеров	65
Технические требования для использования C++.....	65
Изучение основ использования библиотеки C++.....	67
И снова создание массива структур.....	69
Упражнение для самостоятельного выполнения.....	72
Arrow и Go – начинаем	72
И снова создаем массив структур.....	75
Упражнение для самостоятельного выполнения (да, опять то же самое)	77
Резюме.....	77
Ссылки на материалы и ресурсы, упоминаемые в главе 1	77

Глава 2. Использование основных

спецификаций Arrow 79

Технические требования.....	79
Работа с данными независимо от места их расположения	80
Работа с таблицами Arrow.....	81
Доступ к файлам данных с использованием PyArrow	83
Работа с CSV-файлами в PyArrow.....	87
Работа с файлами JSON в PyArrow.....	91
Работа с файлами ORC в PyArrow.....	92
Работа с файлами Apache Parquet в PyArrow	93
Доступ к файлам данных с использованием Arrow на языке C++.....	94
Работа с файлами данных CSV с использованием Arrow на языке C++	94
Работа с файлами данных JSON с использованием Arrow на языке C++	99
Работа с файлами данных ORC с использованием Arrow на языке C++	99
Работа с файлами данных Parquet с использованием Arrow на языке C++	101
Панды запускают Arrow	103
Добавление pandas в личный арсенал	103
Ускоряем работу pandas	107
Сохранение контроля над pandas.....	109
Белые медведи используют «стрелы с ржавчиной»	112
Делиться – значит заботиться... особенно если это ваша память.....	114
Управление памятью.....	114
C++	115
Python.....	116
Golang.....	117
Управление буферами для повышения производительности	118
Пересекая границы.....	120
Резюме.....	123

Глава 3. Формат и работа с оперативной памятью 125

Технические требования	126
Сравнение форматов хранения, форматов представления в памяти во время выполнения и форматов передачи сообщений	126
Форматы долговременного хранения.....	126
Форматы данных в памяти во время выполнения	129
Форматы передачи сообщений	132
Подведем краткие итоги.....	133
Передаем данные Arrow во всех направлениях.....	133
Что же это за волшебство?!.....	134
Производство и потребление данных Arrow	138
Изучение картографии памяти	141
Базовый вариант.....	142
Сравнение Parquet и CSV	143
Отображение данных в память	144
Компьютеры — это магия, или TL;DR	148
Виртуальное и физическое пространство памяти.....	149
Что именно делает отображение в память	150
Отображение в память не самое идеальное решение	152
Оставим в покое ЦП – воспользуемся памятью устройства.....	154
Начнем с нескольких указателей	155
Работа с буфером независимо от конкретного устройства	156
Резюме.....	157

**РАЗДЕЛ 2. ВЗАИМОДЕЙСТВИЕ С ARROW:
МОЩЬ ОТКРЫТЫХ СТАНДАРТОВ 159****Глава 4. Расширение возможностей языков
программирования с помощью Arrow C Data API..... 161**

Технические требования	162
Использование интерфейса Arrow C Data.....	162
Структура ArrowSchema.....	164
Формат типа данных	165
Схема кодирования метаданных	168
О типе расширения	169
Структура ArrowArray	170
Примеры вариантов использования.....	172
Использование C Data API для экспорта данных в формате Arrow	172
Импорт данных Arrow с использованием Python	175
Экспорт данных Arrow с использованием C Data API из Python в Go.....	176
Создание совместно используемой библиотеки на языке Go.....	177
Создание расширения Python из библиотеки Go.....	179
Организация потока данных через C Data API.....	181
Кое-что о данных на устройстве, отличающемся от ЦП.....	185
Структура ArrowDeviceArray	185
Использование структуры ArrowDeviceArray	188

Другие варианты использования	192
Несколько упражнений	193
Резюме	194

Глава 5. Асего: механизм потоковой обработки Arrow 195

Технические требования	196
Пусть Асего работает за вас	196
Формирование входных данных	196
Приведение типа значения	197
Типы функций в Асего	197
Агрегации	197
Поэлементные или скалярные функции	198
Векторные функции для работы с массивом	198
Вызов функций	199
Использование вычислительной библиотеки C++	199
Пример 1 – сложение скалярного значения с массивом	199
Пример 2 – агрегирующая функция <code>min_max</code>	202
Пример 3 – сортировка таблицы данных	202
Использование вычислительной библиотеки в коде Python	203
Пример 1 – сложение скалярного значения с массивом	203
Пример 2 – агрегирующая функция <code>min_max</code>	204
Пример 3 – сортировка таблицы данных	204
Выбор правильных инструментов	205
Сложение значения-константы с массивом	205
Применение функции <code>Add</code> для вычислений	206
Простой цикл <code>for</code>	207
Использование цикла <code>std::for_each</code> с резервированием памяти	207
Методика «разделяй и властвуй»	208
План нужен всегда	211
Где применять Асего?	211
Основные концепции Асего	212
<code>ExecNode</code>	213
<code>ExecBatch</code>	214
<code>ExecPlan</code>	215
<code>Declaration</code>	215
Начинаем обработку потоков	215
Упрощение сложности	218
Резюме	222

Глава 6. Использование интерфейсов Arrow Datasets API 224

Технические требования	225
Выполнение запросов к многофайловым наборам данных	225
Создание примера набора данных	227
Обнаружение фрагментов набора данных	229
Программная фильтрация данных	231
Самовыражение – краткий обзор	231
Использование выражений для фильтрации данных	233

Формирование новых и переименование существующих столбцов (проецирование)	234
Использование Datasets API в коде Python	236
Создание примера набора данных	236
Обнаружение набора данных	237
Использование различных форматов файлов	238
Фильтрация и проецирование столбцов на языке Python	238
Организация потока результатов	239
Работа с разделенными наборами данных	244
Запись разделяемых данных	248
Соединяем все вместе	252
Резюме	253

Глава 7. Изучение протокола Apache Arrow Flight RPC..... 254

Технические требования	255
Основы и сложности gRPC	255
Создание современных API для работы с данными	255
Важна эффективность и возможность потоковой обработки	257
Форматы сообщений	257
Генерация кода	258
Использование и поддержка сети	258
Структурные компоненты Arrow Flight	259
Горизонтальная масштабируемость в Arrow Flight	261
Добавление бизнес-логики клиента в Flight	263
Дополнительные компоненты	264
Изучение определений Flight Protobuf	265
Выберите язык для использования Flight	269
Создание сервера Flight на языке Python	269
Создание сервера Flight на языке Go	272
Первая версия (общая схема) сервера	272
Реализация метода ListFlights	274
Реализация метода DoGet	277
Что такое Flight SQL	280
Настройка теста производительности	281
Каждому разработчику – отдельную среду разработки в контейнере	281
Выполнение теста производительности	285
Новый персонаж Flight SQL	287
Резюме	288

Глава 8. Изучение Arrow Database Connectivity (ADBS) 289

Технические требования	290
Arrow: «стрела в колено» ODBC	290
«Трудности перевода»	291
Внедрение Arrow в драйверы ODBC	292
Преимущества стандартов для организации соединений	293
Спецификация ADBC	296
Базы данных ADBC	297
Соединения ADBC	298

Инструкции ADBC.....	300
Обработка ошибок в ADBC.....	303
Использование ADBC для повышения производительности и адаптируемости.....	303
ADBC и C/C++	303
Простой пример использования ADBC	305
Использование ADBC Driver Manager.....	307
Использование ADBC совместно с Python	310
Массовое потребление данных с помощью ADBC	313
И последнее – организация потока между базами данных.....	315
Использование ADBC совместно с Go	316
Приводим наших «уток» в порядок.....	318
И последнее – использование database/sql.....	321
Резюме.....	322

Глава 9. Использование Arrow в рабочих процессах машинного обучения 324

Технические требования.....	325
Spark как источник новых идей для Jupyter	325
Более подробно об интеграции Arrow в Spark.....	325
Контейнеризация упрощает жизнь.....	327
Искреннее восхищение связкой Arrow и PySpark	329
Шаг 1 – делаем данные пригодными для использования	329
Шаг 2 – добавление нового столбца	335
Шаг 3 – создание UDF для нормализации столбца.....	336
Шаг 4 – отбрасываем шаг 3 и используем векторизованную версию UDF	338
Шаг 5 – подробное объяснение произошедшего.....	339
Шаг 6 – извлечение пользы из нашей напряженной работы	340
Hugging Face: «имплантация» данных	341
Настройка рабочей среды	341
Доказательство преимуществ посредством проверки использования ресурсов	343
Использование Arrow совместно со стандартными инструментами машинного обучения.....	344
Чем больше GPU, тем выше скорость	346
Резюме.....	349

РАЗДЕЛ 3. РЕАЛЬНЫЕ ПРИМЕРЫ, ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ И ПЕРСПЕКТИВЫ ДАЛЬНЕЙШЕГО РАЗВИТИЯ..... 351

Глава 10. Разработки на основе технологии Apache Arrow 353

Навигация по данным с помощью Dremio Sonar	354
Архитектура Dremio Sonar	354
Библиотека Богов... анализа данных	356

«Оживление» рабочих процессов машинного обучения	358
Arrow в браузере, использующем JavaScript	359
Немного о перспективах.....	360
Полет с Falcon.....	361
Приток соединений	362
Резюме.....	363

Глава 11. Как принять участие в разработке Arrow 365

Технические требования	366
Вклад в проекты с открытым исходным кодом	366
Обмен информацией – это главное	366
Необязательно вносить вклад в исходный код	367
Множество причин, по которым вы должны вносить вклад.....	367
Подготовка к первому запросу на включение изменений	369
Создание задачи в GitHub и средства навигации по задачам.....	369
Настройка Git	371
Ориентация в кодовой базе	372
Создание библиотек Arrow	374
Создание библиотек Arrow для C++.....	374
Технические требования для сборки библиотек C++	374
Собираем библиотеку	376
Конфигурирование сборки вручную.....	378
Создание модуля Arrow Python	381
Архитектура ruarrow – Cython – это не просто остроумное имя	382
И снова сборка модуля.....	384
Сборка модулей Arrow Go	385
Создание запроса на внесение изменений (PR).....	387
Изучение Archery и конфигурации непрерывной интеграции (CI).....	388
Автоматизированные задания, активизируемые	
некоторым действием.....	388
Разработка с использованием Archery	389
Выполнение локальных сборок с использованием Archery и Docker	390
Найдите «свою» тему и активно участвуйте в ее разработке.....	391
Получите долгожданное подтверждение.....	392
И в заключение о стиле	392
Стиль кодирования C++.....	393
Стиль кодирования Python	393
Стиль кодирования Go	393
Резюме.....	394

Глава 12. Дальнейшие перспективы и планы разработки..... 395

По всему миру с данными – GeoArrow и GeoParquet	396
Сотрудничество порождает успех	398
Расширение практического применения ADBC.....	399
Заключительное слово	400

Предметный указатель 402

Предисловие от издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в основном тексте или программном коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Вступительное слово

После старта проекта с открытым исходным кодом в 2016 г. Apache Arrow быстро стал фактическим стандартом взаимодействия и ускоренной обработки табличных данных в памяти. Мы расширили проект для поддержки десятка языков программирования, далеко выйдя за пределы его первоначальной цели по определению стандартизированного формата данных, организованных по столбцам, чтобы создать настоящий многоязычный набор инструментов разработчика для создания высокопроизводительных приложений для обработки данных. Хотя Arrow помог существенно улучшить взаимодействие и производительность в гетерогенных системах (например, при использовании разных языков программирования или различных типов механизмов выполнения), тем не менее его все чаще выбирают в качестве основы для создания новых систем обработки данных и баз данных. Учитывая, что Dremio является первой по-настоящему «нативной системой Arrow», мы надеемся, что в ближайшие годы многие другие производственные системы станут «совместимыми с Arrow» или «специально предназначенными для Arrow».

Успех Arrow и быстрый рост сообщества разработчиков в определенной степени обусловлены энтузиазмом и затратами времени первых пользователей и наиболее активных основных участников проекта. Мэтт Топол был главной персоной в разработке библиотек языка Go для Arrow, а этой новой книгой он внес значительный вклад, позволяющий сделать весь проект намного более доступным для новичков. В книге подробно описано, как работают различные части Arrow, и в то же время выделено множество разнообразных компонентов, которые пользователь Arrow может применять для ускорения или упрощения своего приложения.

Я очень рад увидеть обновленное второе издание этой книги, поскольку сам проект Arrow и его экосистема с открытым исходным кодом продолжают расширяться в новых, эффективных направлениях, даже с учетом того факта, что проект существует уже более восьми лет. Это первая настоящая «книга об Arrow» с момента основания проекта, и она является ценным ресурсом для разработчиков, желающих глубоко изучить различные аспекты использования Arrow и научиться применять новые инструменты в своих проектах. Рекомендую ее всем новым и действующим пользователям Arrow, которые хотят углубить свои знания, обучаясь у такого опытного эксперта, как Мэтт.

— Уэс МакКинни (Wes McKinney),
сооснователь Voltron Data и главный архитектор в компании Posit,
соавтор и управляющий проектом по контракту (PMC) Apache Arrow

Об авторе

Мэтью Топол (Matthew Topol) – член комиссии по управлению проектом (PMC) Apache Arrow и ведущий инженер по программному обеспечению в компании Voltron Data. Мэтт работал в сфере разработки инфраструктуры и приложений и проектировал крупномасштабные распределенные системы для обработки и анализа финансовых данных. В компании Voltron Data главная обязанность Мэтта – работа по расширению библиотек Apache Arrow и связанных с ними дочерних проектов. В свободное время Мэтт любит уткнуться в клавиатуру, чтобы разрабатывать и запускать восхитительно безумные игры фэнтези для своих жертв... мы хотели сказать, друзей, и делиться своими знаниями со всеми, кто достаточно заинтересован, чтобы слушать его.

Особая благодарность моим друзьям Хоупу и Стэну, чья поддержка – единственная причина, по которой я вообще написал книгу. И конечно же, спасибо моим родителям, которые сияют от гордости каждый раз, когда я говорю об этой книге. Спасибо вам за вашу поддержку и за то, что вы всегда были рядом.

О рецензентах

Уэстон Пейс (Weston Pace) – разработчик проекта Apache Arrow и член Arrow PMC и Substrait SMC. Он активно работал над реализациями Apache Arrow на языках C++, Python и Rust. Уэстон разработал компоненты для нескольких систем, описанных в этой книге, таких как наборы данных и Acero. В настоящее время Уэстон является сотрудником компании LanceDB, где он работает над новыми форматами хранения, совместимыми с Arrow, чтобы обеспечить возможность еще более широкого применения технологии Arrow.

Джейкоб Вуйцяк-Йенс (Jacob Wujciak-Jens) – коммиттер Apache Arrow и избранный член Apache Software Foundation. Его работа в Voltron Data в качестве старшего инженера по выпуску программного обеспечения включала ключевые роли в проектах Apache Arrow и Velox. За время своей работы он приобрел глубокие знания о процессах выпуска, системах сборки и внутренних механизмах этих широко известных проектов с открытым исходным кодом. Джейкоб является активным сторонником программного обеспечения с открытым исходным кодом и его использованием как в сообществе, так и в производственных отраслях. Имея степень магистра педагогических наук в области информатики и общественного здравоохранения, Джейкоб любит делиться своими знаниями, обогащая сообщество и улучшая совместные проекты.

Рауль Кумплидо (Raúl Cumplido) – член комиссии по управлению проектом (PMC) Apache Arrow, менеджер по выпуску более 10 релизов проекта. Он работал над несколькими направлениями проекта Arrow. Рауль всегда был связан с сообществами с открытым исходным кодом, внося вклад в основном в проекты, связанные с Python. Рауль является одним из соучредителей Python Spanish Association, а кроме того, одним из организаторов нескольких конференций EuroPython и PyCon ES. В настоящее время Рауль работает старшим инженером по выпуску программного обеспечения в Voltron Data, где принимает участие в проекте Apache Arrow.

Предисловие

Как сказал известный синий ежик (Соник), «Gotta Go Fast!» (Пора ускоряться!) Когда дело касается данных, важна скорость. Неважно, собираете ли вы или анализируете данные или разрабатываете для кого-то утилиты, производительность и эффективность всегда являются важнейшими факторами при выборе технологий, причем это относится не только к эффективности самого программного обеспечения, но и ко времени разработки. Вам необходимы правильные инструменты и правильная технология, или вы увязнете в болоте.

Экосистема Apache Arrow ориентирована на разработчиков, и эта книга тоже. Начните с изучения того, что такое Arrow и как работает эта система, а затем узнайте, как использовать ее в своих проектах. Здесь вы найдете примеры кода, пояснения и схемы – все с очевидной целью помочь вам в обучении. Вы научитесь интегрировать источники данных с библиотеками Python DataFrame, такими как pandas или NumPy, и использовать Arrow Flight для создания эффективных сервисов обработки данных.

Вы узнаете, как использовать Apache Arrow совместно с Apache Spark и другими технологиями для работы с реальными наборами данных. Формат Apache Arrow не зависит от какого-либо языка и организован так, что аналитические операции выполняются чрезвычайно быстро на современных центральных и графических процессорах. Присоединяйтесь к движению внедрения в промышленном масштабе этого формата данных с открытым исходным кодом и сэкономьте драгоценное время разработки, создавая высокопроизводительные, эффективные в использовании памяти аналитические рабочие процессы.

Эта книга появилась как результат постоянной потребности делиться знаниями. Надеюсь, вы многому научитесь из нее. Даже когда я писал эту книгу, тоже многому научился.

Для кого предназначена эта книга

Эта книга предназначена для разработчиков, аналитиков данных и специалистов по обработке данных, которые хотят изучить возможности Apache Arrow с нуля. Она также будет полезна для всех инженеров, которые проектируют и создают утилиты для анализа данных, механизмы запросов или какими-ли-

бо иными способами работают с табличными данными, независимо от применяемого языка программирования.

Темы этой книги

Глава 1 «Начинаем работу с Apache Arrow» представляет концепции, заложенные в основу Apache Arrow. Здесь описывается формат Arrow и поддерживаемые типы данных, а также их представление в оперативной памяти. Далее рассматривается настройка среды разработки и выполнение нескольких простых примеров кода, демонстрирующих основные операции библиотек Arrow.

Глава 2 «Использование основных спецификаций Arrow» продолжает знакомить вас с Apache Arrow, объясняя, как считывать локальные и удаленные файлы данных в различных форматах. Вы узнаете, как интегрировать Arrow с библиотеками Python pandas и Polaris и как применять приемы нулевого копирования Arrow при совместном использовании оперативной памяти для повышения производительности.

В главе 3 «Формат и работа с оперативной памятью» рассматриваются взаимоотношения между Apache Arrow и данными в форматах Apache Parquet, Feather, Protocol Buffers, JSON и CSV, а также объясняется, когда и почему применяются эти разнообразные форматы. Далее описывается формат Arrow IPC и практическое использование его отображения в оперативной памяти для обеспечения еще большей производительности.

Глава 4 «Расширение возможностей языков программирования с помощью Arrow C Data API» представляет официальный интерфейс C Data API, предназначенный для эффективной передачи данных Apache Arrow между средами времени выполнения различных языков программирования и разнообразными устройствами. В этой главе подробно рассматриваются определения структур, используемых для этого интерфейса, а также описываются варианты использования, которые делают его наиболее предпочтительным вариантом выбора.

В главе 5 «Acero: механизм потоковой обработки Arrow» описывается применение на практике базовой реализации вычислительного механизма Arrow под названием Acero. Вы узнаете, когда и почему следует использовать этот механизм для анализа вместо реализации собственных инструментов, и почему мы обнаруживаем Arrow во многих широко распространенных механизмах выполнения.

В главе 6 «Использование интерфейсов Arrow Datasets API» демонстрируется выполнение запросов, фильтрации и других методов взаимодействия с наборами данных, размещенных в нескольких файлах и, возможно, собранных из многочисленных разнообразных источников. Также рассматриваются наборы данных, разделенные на части, и применение Acero для выполнения потоковой обработки с фильтрацией и других операций с данными.

В главе 7 «Изучение протокола Apache Arrow Flight RPC» представлено описание протокола Flight RPC и преимуществ его использования. Подробно рассматривается пошаговая процедура создания простого сервера Flight и соответствующего клиента на нескольких языках программирования для производства и потребления табличных данных.

В главе 8 «Изучение Arrow Database Connectivity (ADBC)» рассматривается вариант замены протоколов ODBC/JDBC механизмом на основе Apache Arrow и объясняется, почему это важно для всей экосистемы в целом. Подробно разбираются примеры исходного кода, взаимодействующие с несколькими системами управления базами данных, такими как DuckDB и PostgreSQL.

Глава 9 «Использование Arrow в рабочих процессах машинного обучения» объединяет несколько концепций, которые рассматривались при описании разнообразных способов применения Apache Arrow для улучшения качества частей конвейера данных и повышения производительности тренировки моделей машинного обучения. Показано, как универсальность и определенные стандарты Arrow делают этот фреймворк идеальным для совместного использования со Spark, для вычислений с применением GPU и многих других инструментальных средств.

В главе 10 «Разработки на основе технологии Apache Arrow» приведено несколько примеров современного использования Arrow в реальной производственной среде, такой как Dremio, Spice.ai и InfluxDB.

Глава 11 «Как принять участие в разработке Arrow» предлагает краткое введение для тех, кто желает внести свой вклад в разработку программного обеспечения с открытым исходным кодом вообще и в частности в разработку проекта Arrow. Приведена подробная инструкция по поиску стартующих проектов и задач, а также по формулированию первого запроса на включение внесенных изменений, позволяющего принять участие в проекте, и объясняется, чего следует ожидать при этом. Глава также содержит различные инструкции по локальной сборке библиотек Arrow C++, Python и Go, чтобы проверить вашу готовность к участию.

Глава 12 «Дальнейшие перспективы и планы разработки» завершает книгу описанием функциональных средств и возможностей, разработка которых продолжается во время написания этой книги, включая геопространственные интеграции с GeoArrow и GeoParquet, а также расширение адаптации Arrow Database Connectivity (ADBC). Завершается глава прощальными пожеланиями и рекомендациями автора читателям.

КАК ИЗВЛЕЧЬ МАКСИМАЛЬНУЮ ПОЛЬЗУ ИЗ ЭТОЙ КНИГИ

Предполагается, что читатель обладает базовыми знаниями о написании кода по крайней мере на одном из языков C++, Python или Go, чтобы использовать фрагменты кода. Читатель должен знать, как компилировать и запускать код на конкретном языке, применяемом в примерах. Некоторое знакомство с базовыми концепциями анализа данных поможет вам извлечь максимальную пользу из этой книги. Кроме того, предполагается, что читатель понимает такие концепции, как табличные данные и установка программного обеспечения на компьютере.

Выборки данных размещены в репозитории GitHub этой книги. Потребуется использование Git Large File Storage (LFS) или браузера для загрузки файлов больших данных. Кроме того, несколько файлов с выборками больших данных находятся в общедоступном контейнере AWS S3. При необходимости в книге приводятся ссылки для загрузки этих файлов. Исходные коды примеров представлены на языках C++, Python и Go.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru