УДК 004.43Spark
ББК 32.972
П26

Перрен Ж.-Ж.
П26 Spark в действии / пер. с анг. А. В. Снастина. – М.: ДМК Пресс, 2021. – 636 с.: ил.

ISBN 978-5-97060-879-1

Обработка больших данных с каждым днем приобретает все большее значение. В этой книге подробно рассматривается организация обработки больших данных с использованием аналитической операционной системы Apache Spark. Тщательно описываются процессы потребления, преобразования и публикации результатов обработки данных; продемонстрированы возможности Apache Spark при работе с разнообразными форматами исходных данных (текст, JSON, XML, СУРБД и многими другими) и при публикации результатов в разнообразных форматах. Особое внимание уделяется обработке потоковых данных, что весьма важно в современных условиях. Подробно рассмотрены организация и архитектура кластера Spark. В приложениях представлена обширная справочная информация, необходимая каждому разработчику, использующему Spark.

Книга содержит множество иллюстраций и примеров исходного кода на языке Java с подробными комментариями.

Издание предназначено для разработчиков, начинающих осваивать систему Spark.

УДК 004.43Spark
ББК 32.972

ISBN 978-1-6172-9552-2 (анг.)
ISBN 978-5-97060-879-1 (рус.)

Содержание

Оглавление 5
Словарь терминов 15
Вступительное слово 17
Предисловие 19
Благодарности 21
О чем эта книга 24
Об авторе 32
Иллюстрация на обложке 33

Часть I Теория, разбавленная превосходными примерами 35

1 Так что же такое Spark? 36

1.1 Общая картина: что такое Spark и что он делает 37
1.1.1 Что такое Spark 37
1.1.2 Четыре столпа маны 40
1.2 Как можно использовать Spark 41
1.2.1 Spark в процессе обработки данных / инженерии данных 41
1.2.2 Spark в научных исследованиях в области обработки данных 44
1.3 Что можно делать с помощью Spark 45
1.3.1 Spark прогнозирует качество пунктов питания Северной Каролины 46
1.3.2 Spark обеспечивает быструю передачу данных для Lumeris 47
1.3.3 Spark анализирует журналы наблюдения за оборудованием CERN 48
1.3.4 Другие варианты использования 48

1.4 Почему вам очень понравится фрейм данных 48

1.4.1 Фрейм данных с точки зрения Java 49

1.4.2 Фрейм данных с точки зрения СУРБД 49

1.4.3 Графическое представление фрейма данных 50

1.5 Первый пример 51

1.5.1 Рекомендуемое программное обеспечение 51

1.5.2 Скачивание исходного кода 52

1.5.3 Запуск первого приложения 52

1.5.4 Первый исходный код для вас 53

Резюме 54

2 *Архитектура и рабочий процесс* 56

2.1 Создание собственной мысленной (когнитивной) модели 57

2.2 Использование кода Java для создания мысленной (когнитивной) модели 58

2.3 Подробный разбор приложения 61

2.3.1 Установление соединения с ведущим узлом 62

2.3.2 Загрузка или потребление содержимого CSV-файла 63

2.3.3 Преобразование данных 66

2.3.4 Сохранение работы, сделанной в фрейме данных, в базе данных 68

Резюме 71

3 *Важнейшая роль фрейма данных* 72

3.1 Чрезвычайно важная роль фрейма данных в Spark 73

3.1.1 Внутренняя организация фрейма данных 74

3.1.2 Неизменяемость – это не клятва 75

3.2 Использование фреймов данных на примерах 77

3.2.1 Фрейм данных после простой операции потребления CSV-файла 79

3.2.2 Данные хранятся в разделах 84

3.2.3 Подробнее о схеме 86

3.2.4 Фрейм данных после потребления формата JSON 87

3.2.5 Объединение двух фреймов данных 94

3.3 Фрейм данных как структура `Dataset<Row>` 99

3.3.1 Повторное использование простых старых объектов Java (POJO) 100

3.3.2 Создание набора данных из строк 101

3.3.3 Преобразование фрейма данных в набор данных и обратно 103

3.4 Предшественник фрейма данных: RDD 109

Резюме 110

4 *Природная лень* 112
4.1 Пример рациональной лени из реальной жизни 113
4.2 Пример рациональной лени в Spark 114
4.2.1 Рассмотрение результатов преобразований и действий 115
4.2.2 Процесс преобразования шаг за шагом 116
4.2.3 Код реализации процесса преобразования/действия 119
4.2.4 Загадка создания 7 миллионов точек данных за 182 мс 123
4.2.5 Загадка, связанная с измерением времени для действий 125
4.3 Сравнение с СУРБД и обычными приложениями 130
4.3.1 Обработка набора данных с коэффициентами рождаемости для подростков 130
4.3.2 Анализ различий между обычным приложением и приложением Spark 131
4.4 Spark великолепно подходит для приложений, ориентированных на обработку данных 133
4.5 Catalyst – катализатор приложения 133
Резюме 137

5 *Создание простого приложения для развертывания* 138
5.1 Пример без операции потребления данных 139
5.1.1 Вычисление π 139
5.1.2 Исходный код для вычисления приближенного значения π 142
5.1.3 Что такое лямбда-функции в Java 148
5.1.4 Приближенное вычисление π с использованием лямбда-функций 150
5.2 Взаимодействие со Spark 152
5.2.1 Локальный режим 153
5.2.2 Режим кластера 154
5.2.3 Интерактивный режим Scala и Python 158
Резюме 163

6 *Развертывание простого приложения* 165
6.1 Подготовка к изучению примера: роль компонент 168
6.1.1 Краткий обзор компонент и взаимодействий между ними 168
6.1.2 Рекомендации по устранению проблем в архитектуре Spark 172
6.1.3 Дополнительная информация для изучения 173
6.2 Создание кластера 174
6.2.1 Создание собственного кластера 174
6.2.2 Настройка среды кластера 176

6.3 Создание приложения для работы в кластере 179

6.3.1 Создание файла uberJAR для приложения 180

6.3.2 Создание приложения с использованием Git и Maven 182

6.4 Выполнение приложения в кластере 185

6.4.1 Передача файла uberJAR 185

6.4.2 Выполнение приложения 186

6.4.3 Анализ пользовательского интерфейса Spark 187

Резюме 188

Часть II Потребление данных 190

7 *Потребление данных из файлов* 192

7.1 Общее поведение парсеров 194

7.2 Сложная процедура потребления данных из CSV-файла 194

7.2.1 Требуемый вывод результата 196

7.2.2 Код 197

7.3 Потребление CSV-данных с известной схемой 198

7.3.1 Требуемый вывод результата 199

7.3.2 Код 200

7.4 Потребление данных из JSON-файла 201

7.4.1 Требуемый вывод результата 203

7.4.2 Код 204

7.5 Потребление данных из многострочного JSON-файла 205

7.5.1 Требуемый вывод результата 207

7.5.2 Код 207

7.6 Потребление данных из файла XML 208

7.6.1 Требуемый вывод результата 210

7.6.2 Код 211

7.7 Потребление данных из текстового файла 213

7.7.1 Требуемый вывод результата 214

7.7.2 Код 214

7.8 Форматы файлов для больших данных 215

7.8.1 Проблема с обычными форматами файлов 215

7.8.2 Avro – формат сериализации на основе схемы 217

7.8.3 ORC – формат хранения данных в столбцах 217

7.8.4 Parquet – еще один формат хранения данных в столбцах 218

7.8.5 Сравнение форматов Avro, ORC и Parquet 218

7.9 Потребление данных из файлов Avro, ORC и Parquet 218

7.9.1 Потребление данных в формате Avro 219

7.9.2 Потребление данных в формате ORC 221

7.9.3 Потребление данных в формате Parquet 222

7.9.4 Справочная информация по организации потребления данных в форматах Avro, ORC, Parquet 224

Резюме 224

8 *Потребление из баз данных* ... 226

8.1 Потребление из реляционных баз данных ... 228

8.1.1 Контрольный перечень операций при установлении соединения с базой данных ... 228

8.1.2 Объяснение происхождения данных, используемых в следующих примерах ... 229

8.1.3 Требуемый вывод результата ... 231

8.1.4 Код ... 232

8.1.5 Другая версия кода ... 234

8.2 Роль диалекта ... 236

8.2.1 Что такое диалект ... 236

8.2.2 Диалекты JDBC, предоставляемые в Spark ... 237

8.2.3 Создание собственного диалекта ... 237

8.3 Расширенные запросы и процесс потребления ... 240

8.3.1 Фильтрация с использованием ключевого слова `WHERE` ... 240

8.3.2 Соединение данных в базе данных ... 243

8.3.3 Выполнение потребления и распределение данных ... 246

8.3.4 Итоги изучения расширенных функциональных возможностей ... 249

8.4 Потребление данных из Elasticsearch ... 249

8.4.1 Поток данных ... 249

8.4.2 Набор данных о ресторанах Нью-Йорка, извлекаемый Spark ... 250

8.4.3 Исходный код для потребления набора данных о ресторанах из Elasticsearch ... 252

Резюме ... 253

9 *Более сложный процесс потребления: поиск источников данных и создание собственных* ... 255

9.1 Что такое источник данных ... 257

9.2 Преимущества прямого соединения с источником данных ... 259

9.2.1 Временные файлы ... 260

9.2.2 Скрипты для улучшения качества данных ... 260

9.2.3 Данные по запросу ... 261

9.3 Поиск источников данных на сайте Spark Packages ... 261

9.4 Создание собственного источника данных ... 261

9.4.1 Обзор примера проекта ... 262

9.4.2 Интерфейс API специализированного источника данных и его параметры ... 264

9.5 Что происходит внутри: создание самого источника данных ... 267

9.6 Использование файла регистрации и заявочного класса ... 268

9.7 Объяснение взаимоотношения между данными и схемой ... 270

9.7.1 Источник данных создает отношение ... 271

9.7.2 Внутри отношения ... 274

9.8 Создание схемы из JavaBean ... 277

9.9 Создание фрейма данных – манипуляции с утилитами ... 280

9.10 Другие классы ... 286

Резюме ... 286

10 *Потребление через структурированные потоки* ... 288

10.1 Что такое потоковая обработка ... 290

10.2 Создание первого потока данных ... 292

10.2.1 Генерация потока данных ... 293

10.2.2 Потребление записей ... 296

10.2.3 Считывание записей, а не строк ... 302

10.3 Потребление данных из сетевых потоков ... 303

10.4 Работа с несколькими потоками ... 306

10.5 Различия между дискретизированными и структурированными потоками ... 311

Резюме ... 312

Часть III Преобразование данных ... 313

11 *Работа с языком SQL* ... 314

11.1 Работа со Spark SQL ... 315

11.2 Различия между локальными и глобальными представлениями ... 319

11.3 Совместное использование API фрейма данных и Spark SQL ... 321

11.4 Не удаляйте (`DELETE`) данные ... 324

11.5 Рекомендации для дальнейшего изучения SQL ... 327

Резюме ... 327

12 *Преобразование данных* ... 329

12.1 Что такое преобразование данных ... 330

12.2 Процесс и пример преобразования данных на уровне записи ... 331

12.2.1 Обследование данных для оценки их сложности ... 333

12.2.2 Отображение данных для создания схемы процесса ... 335

12.2.3 Написание исходного кода преобразования ... 338

12.2.4 Итоговый обзор результата преобразования данных для обеспечения качества обработки ... 345

12.2.5 Несколько слов о сортировке ... 347

12.2.6 *Завершение первого процесса преобразования с использованием Spark* 347

12.3 Соединение наборов данных 348

12.3.1 *Более подробно о соединяемых наборах данных* 348

12.3.2 *Создание списка вузов по округам* 350

12.3.3 *Выполнение соединений* 356

12.4 Выполнение других преобразований 362

Резюме 362

13 ***Преобразование документов в целом*** 364

13.1 Преобразование документов в целом и их структура 365

13.1.1 *Упрощение структуры документа в формате JSON* 365

13.1.2 *Создание документов с вложенной структурой для передачи и сохранения* 371

13.2 Секреты статических функций 376

13.3 Выполнение других преобразований 377

Резюме 377

14 ***Расширенные преобразования с помощью функций, определенных пользователем*** 378

14.1 Расширение функциональности Apache Spark 379

14.2 Регистрация и вызов UDF 381

14.2.1 *Регистрация UDF в Spark* 384

14.2.2 *Использование UDF совместно с API фрейма данных* 385

14.2.3 *Использование UDF совместно с SQL* 387

14.2.4 *Реализация UDF* 388

14.2.5 *Написание кода сервиса* 390

14.3 Использование UDF для обеспечения высокого уровня качества данных 392

14.4 Ограничения использования UDF 394

Резюме 395

15 ***Агрегирование данных*** 396

15.1 Агрегирование данных в Spark 397

15.1.1 *Краткое описание агрегаций* 397

15.1.2 *Выполнение простых агрегаций с использованием Spark* 400

15.2 Выполнение агрегаций с оперативными данными 403

15.2.1 *Подготовка набора данных* 403

15.2.2 *Агрегация данных для получения более точной информации о школах* 408

15.3 Создание специализированных агрегаций с использованием UDAF 415

Резюме 422

Часть IV Продолжаем изучение Spark 424

16 *Кеширование и копирование данных в контрольных точках: улучшение производительности Spark* 426

16.1 Кеширование и копирование данных в контрольных точках могут повысить производительность 427

16.1.1 Полезность кеширования в Spark 429

16.1.2 Изысканная эффективность механизма копирования данных в контрольных точках в Spark 431

16.1.3 Использование кеширования и копирования данных в контрольных точках 431

16.2 Кеширование на практике 442

16.3 Дополнительные материалы по оптимизации производительности 452

Резюме 453

17 *Экспорт данных и создание полноценных конвейеров обработки данных* 455

17.1 Экспорт данных 456

17.1.1 Создание конвейера с наборами данных NASA 456

17.1.2. Преобразование столбцов в метки времени datetime 459

17.1.3 Преобразование процентов степени достоверности в уровень достоверности 461

17.1.4 Экспорт данных 462

17.1.5 Экспорт данных: что происходит в действительности 465

17.2 Delta Lake: удобная база данных прямо в системе 466

17.2.1 Объяснение, почему необходима база данных 467

17.2.2 Использование Delta Lake в конвейере обработки данных 468

17.2.3 Потребление данных из Delta Lake 473

17.3 Доступ к сервисам облачного хранилища из Spark 475

Резюме 477

18 *Описание ограничений процесса развертывания: объяснение экосистемы* 478

18.1 Управление ресурсами с использованием YARN, Mesos и Kubernetes 479

18.1.1 Встроенный автономный режим управления ресурсами 480

18.1.2 YARN управляет ресурсами в среде Hadoop 481

18.1.3 Mesos – автономный диспетчер ресурсов 483

18.1.4 Kubernetes управляет оркестровкой контейнеров 484

18.1.5 Правильный выбор диспетчера ресурсов 486

18.2 Совместное использование файлов с помощью Spark 486
18.2.1 Доступ к данным, содержащимся в файлах 487
18.2.2 Совместное использование файлов с помощью распределенных файловых систем 488
18.2.3 Доступ к файлам на совместно используемых накопителях или на файловом сервере 490
18.2.4 Работа с сервисами совместного использования файлов для распределения файлов 491
18.2.5 Другие варианты обеспечения доступа к файлам в Spark 492
18.2.6 Гибридное решение совместного использования файлов в Spark 492
18.3 Уверенность в безопасности приложения Spark 492
18.3.1 Безопасность сетевых компонентов инфраструктуры 493
18.3.2 Безопасность при использовании диска Spark 494
Резюме 495

Приложение A Установка Eclipse 496
Приложение B Установка Maven 502
Приложение C Установка Git 506
Приложение D Загрузка исходного кода и начало работы в Eclipse 508
Приложение E Хронология корпоративных данных 514
Приложение F Справочная информация по реляционным базам данных 519
Приложение G Статические функции упрощают преобразования 524
Приложение H Краткий справочник по Maven 533
Приложение I Справочник по преобразованиям и действиям 538
Приложение J Немного Scala 548
Приложение K Установка Spark в реальной эксплуатационной среде и несколько рекомендаций 550
Приложение L Справочник по операциям потребления 563
Приложение M Справочник по соединениям 574
Приложение N Установка Elasticsearch и пример набора данных 586
Приложение O Генерация потоковых данных 592
Приложение P Справочник по обработке потоковых данных 597
Приложение Q Справочник по экспорту данных 608
Приложение R Где искать помощь при затруднениях 616

Предметный указатель 621

ɳqdqɏɘɎ bɆ dɘqɆbɎɜɘ d ɕɑɑbɔɏ Ɉqdqʎqɏè