
Spis treści

Przedmowa	13
Wprowadzenie	15
1. Wprowadzenie do Apache Spark — ujednolicony silnik analityczny	21
Geneza Sparka	21
Big data i przetwarzanie rozproszone w Google	21
Hadoop w Yahoo!	22
Wczesne lata Sparka w AMPLab	23
Czym jest Apache Spark?	24
Szybkość	24
Łatwość użycia	25
Modułowość	25
Rozszerzalność	25
Ujednolicona analityka	25
Komponenty Apache Spark tworzą ujednolicony stos	26
Spark MLlib	27
Wykonywanie rozproszone w Apache Spark	30
Z punktu widzenia programisty	34
Kto używa Sparka i w jakim celu?	34
Popularność w społeczności i dalsza ekspansja	36
2. Pobranie Apache Spark i rozpoczęcie pracy	38
Krok 1. — pobranie Apache Spark	38
Pliki i katalogi Sparka	39
Krok 2. — używanie powłoki Scali lub PySparka	40
Używanie komputera lokalnego	42
Krok 3. — poznanie koncepcji aplikacji Apache Spark	44
Aplikacja Sparka i SparkSession	44
Zlecenia Sparka	45
Etapy Sparka	46
Zadania Sparka	46

Transformacje, akcje i późna ocena	46
Transformacje wąskie i szerokie	48
Spark UI	48
Pierwsza niezależna aplikacja	52
Zliczanie cukierków M&M's	52
Tworzenie niezależnych aplikacji w Scali	57
Podsumowanie	58
3. API strukturalne Apache Spark	59
Spark — co się kryje za akronimem RDD?	59
Strukturyzacja Sparka	60
Kluczowe zalety i wartość struktury	61
API DataFrame	63
Podstawowe typy danych Sparka	64
Strukturalne i złożone typy danych Sparka	65
Schemat i tworzenie egzemplarza DataFrame	65
Kolumny i wyrażenia	69
Rekord	72
Najczęściej przeprowadzane operacje z użyciem DataFrame	73
Przykład pełnego rozwiązania wykorzystującego DataFrame	82
API Dataset	83
Obiekty typowane i nietypowane oraz ogólne rekordy	84
Tworzenie egzemplarza Dataset	85
Operacje na egzemplarzu Dataset	86
Przykład pełnego rozwiązania wykorzystującego Dataset	87
Egzemplarz DataFrame kontra Dataset	88
Kiedy używać RDD?	89
Silnik Spark SQL	90
Optymalizator Catalyst	90
Podsumowanie	95
4. Spark SQL i DataFrame — wprowadzenie do wbudowanych źródeł danych	96
Używanie Spark SQL w aplikacji Sparka	97
Przykłady podstawowych zapytań	97
Widoki i tabele SQL	102
Tabele zarządzane kontra tabele niezarządzane	102
Tworzenie baz danych i tabel SQL	102
Tworzenie widoku	104
Wyświetlanie metadanych	105
Buforowanie tabel SQL	106
Wczytywanie zawartości tabeli do egzemplarza DataFrame	106

Źródła danych dla egzemplarzy DataFrame i tabel SQL	106
DataFrameReader	107
DataFrameWriter	108
Parquet	109
JSON	112
CSV	114
Avro	116
ORC	119
Obrazy	120
Pliki binarne	121
Podsumowanie	123
5. Spark SQL i DataFrame — współpraca z zewnętrznymi źródłami danych	124
Spark SQL i Apache Hive	124
Funkcje zdefiniowane przez użytkownika	125
Wykonywanie zapytań z użyciem powłoki Spark SQL, Beeline i Tableau	129
Używanie powłoki Spark SQL	130
Praca z narzędziem Beeline	131
Praca z Tableau	132
Zewnętrzne źródła danych	138
Bazy danych SQL i JDBC	138
PostgreSQL	140
MySQL	141
Azure Cosmos DB	142
MS SQL Server	144
Inne zewnętrzne źródła danych	145
Funkcje wyższego rzędu w egzemplarzach DataFrame i silniku Spark SQL	146
Opcja 1. — konwersja struktury	146
Opcja 2. — funkcja zdefiniowana przez użytkownika	146
Wbudowane funkcje dla złożonych typów danych	147
Funkcje wyższego rzędu	149
Najczęściej wykonywane operacje w DataFrame i Spark SQL	152
Suma	155
Złączenie	156
Okno czasowe	157
Modyfikacje	159
Podsumowanie	162
6. Spark SQL i Dataset	163
Pojedyncze API dla Javy i Scali	163
Klasy case Scali i JavaBean dla egzemplarzy Dataset	164

Praca z egzemplarzem Dataset	166
Tworzenie przykładowych danych	166
Transformacja przykładowych danych	167
Zarządzanie pamięcią podczas pracy z egzemplarzami Dataset i DataFrame	172
Kodeki egzemplarza Dataset	173
Wewnętrzny format Sparka kontra format obiektu Javy	173
Serializacja i deserializacja	174
Koszt związany z używaniem egzemplarza Dataset	175
Strategie pozwalające obniżyć koszty	175
Podsumowanie	177
7. Optymalizacja i dostrajanie aplikacji Sparka	178
Optymalizacja i dostrajanie Sparka w celu zapewnienia efektywności działania	178
Wyświetlanie i definiowanie konfiguracji Apache Spark	178
Skalowanie Sparka pod kątem ogromnych obciążeń	181
Buforowanie i trwałe przechowywanie danych	187
DataFrame.cache()	187
DataFrame.persist()	188
Kiedy buforować i trwałe przechowywać dane?	191
Kiedy nie buforować i nie przechowywać trwale danych?	191
Rodzina złączeń w Sparku	191
Złączenie BHJ	192
Złączenie SMJ	193
Spark UI	199
Karty narzędzia Spark UI	200
Podsumowanie	207
8. Strumieniowanie strukturalne	208
Ewolucja silnika przetwarzania strumieni w Apache Spark	208
Przetwarzanie strumieniowe mikropartii	209
Cechy mechanizmu Spark Streaming (DStreams)	210
Filozofia strumieniowania strukturalnego	211
Model programowania strumieniowania strukturalnego	211
Podstawy zapytania strumieniowania strukturalnego	214
Pięć kroków do zdefiniowania zapytania strumieniowego	214
Pod maską aktywnego zapytania strumieniowanego	220
Odzyskiwanie danych po awarii i gwarancja „dokładnie raz”	221
Monitorowanie aktywnego zapytania	223
Źródło i ujście strumieniowanych danych	226
Pliki	226
Apache Kafka	228
Niestandardowe źródła strumieni i ujść danych	230

Transformacje danych	234
Wykonywanie przyrostowe i stan strumieniowania	234
Transformacje bezstanowe	235
Transformacje stanowe	235
Agregacje strumieniowania	237
Agregacja nieuwzględniająca czasu	237
Agregacje z oknami czasowymi na podstawie zdarzeń	239
Złączenie strumieniowane	244
Złączenie strumienia i egzemplarza statycznego	245
Złączenia między egzemplarzami strumieniowanymi	246
Dowolne operacje związane ze stanem	251
Modelowanie za pomocą mapGroupsWithState() dowolnych operacji stanu	252
Stosowanie limitów czasu do zarządzania nieaktywnymi grupami	255
Generalizacja z użyciem wywołania flatMapGroupsWithState()	258
Dostrajanie wydajności działania	259
Podsumowanie	261
9. Tworzenie niezawodnych jezior danych za pomocą Apache Spark	262
Waga optymalnego rozwiązania w zakresie pamięci masowej	262
Bazy danych	263
Krótkie wprowadzenie do SQL	263
Odczytywanie i zapisywanie informacji w bazie danych za pomocą Apache Spark	264
Ograniczenia baz danych	264
Jezioro danych	265
Krótkie wprowadzenie do jezior danych	266
Odczytywanie i zapisywanie danych jeziora danych za pomocą Apache Spark	266
Ograniczenia jezior danych	267
Lakehouse — następny krok w ewolucji rozwiązań pamięci masowej	268
Apache Hudi	269
Apache Iceberg	270
Delta Lake	270
Tworzenie repozytorium danych za pomocą Apache Spark i Delta Lake	271
Konfiguracja Apache Spark i Delta Lake	272
Wczytywanie danych do tabeli Delta Lake	272
Wczytywanie strumieni danych do tabeli Delta Lake	274
Zarządzanie schematem podczas zapisu w celu zapobiegania uszkodzeniu danych	275
Ewolucja schematu w celu dostosowania go do zmieniających się danych	276
Transformacja istniejących danych	276
Audyt zmian danych przeprowadzany za pomocą historii operacji	279
Wykonywanie zapytań do poprzednich migawek tabeli dzięki funkcjonalności podróży w czasie	280
Podsumowanie	280

10. Uczenie maszynowe z użyciem biblioteki MLlib	282
Czym jest uczenie maszynowe?	283
Nadzorowane uczenie maszynowe	283
Nienadzorowane uczenie maszynowe	285
Dlaczego Spark dla uczenia maszynowego?	286
Projektowanie potoków uczenia maszynowego	286
Wczytywanie i przygotowywanie danych	287
Tworzenie zbiorów danych — testowego i treningowego	288
Przygotowywanie cech za pomocą transformerów	290
Regresja liniowa	291
Stosowanie estymatorów do tworzenia modeli	292
Tworzenie potoku	293
Ocena modelu	299
Zapisywanie i wczytywanie modeli	303
Dostrajanie hiperparametru	304
Modele oparte na drzewach	304
k-krotny sprawdzian krzyżowy	312
Optymalizacja potoku	315
Podsumowanie	317
 11. Stosowanie Apache Spark do wdrażania potoków uczenia maszynowego oraz ich skalowania i zarządzania nimi	 318
Zarządzanie modelem	318
MLflow	319
Opcje wdrażania modelu za pomocą MLlib	325
Wsadowe	326
Strumieniowane	328
Wzorce eksportu modelu dla rozwiązania niemalże w czasie rzeczywistym	329
Wykorzystanie Sparka do pracy z modelami, które nie zostały utworzone za pomocą MLlib	330
Zdefiniowane przez użytkownika funkcje pandas	330
Spark i rozproszone dostrajanie hiperparametru	332
Podsumowanie	335
 12. Epilog — Apache Spark 3.0	 336
Spark Core i Spark SQL	336
Dynamiczne oczyszczanie partycji	336
Adaptacyjne wykonywanie zapytań	338
Podpowiedzi dotyczące złączeń SQL	341
API wtyczek katalogu i DataSourceV2	342
Planowanie z użyciem akceleratorów	343
Strumieniowanie strukturalne	344

PySpark, zdefiniowane przez użytkownika funkcje pandas i API funkcji pandas	345
Usprawnione zdefiniowane przez użytkownika funkcje pandas	
zapewniające obsługę odpowiedzi typów w Pythonie	346
Obsługa iteratora w zdefiniowanych przez użytkownika funkcjach pandas	347
Nowe API funkcji pandas	348
Zmieniona funkcjonalność	349
Obsługiwane języki	349
Zmiany w API DataFrame i Dataset	349
Polecenia SQL EXPLAIN i DataFrame	350
Podsumowanie	352