Apache Spark Grundlagen

Seminarunterlage

Version: 1.02

Version 1.03 vom
11. Februar 2022

Adressen der ORDIX AG

Die ORDIX AG besitzt folgende Geschäftsstellen

ORDIX AG
Karl-Schurz-Straße 19a
D-33100 Paderborn
Tel.: (+49) 0 52 51 / 10 63 - 0
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Kreuzberger Ring 13
D-65205 Wiesbaden
Tel.: (+49) 06 11 / 7 78 40 – 00
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
An der alten Ziegelei 5
D-48157 Münster
Tel.: (+49) 02 51 / 9 24 35 – 00
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Wikingerstraße 18-20
D-51107 Köln
Tel.: (+49) 02 21 / 8 70 61 – 0
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Welser Straße 9
D-86368 Gersthofen
Tel.: (+49) 08 21 / 507 492 – 0
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Südwestpark 67/2
D-90449 Nürnberg
Tel.: (+49) 0 52 51 / 10 63 - 0
Fax.: (+49) 01 80 / 1 67 34 90

Internet: http://www.ordix.de

Email: seminare@ordix.de

Inhaltsverzeichnis

1 Einführung 7

1.1 Definition 8
1.2 Spark auf einen Blick 11
1.3 Spark Geschichte 12
1.4 Spark Anwendungsfälle 13
1.5 Spark Performance 14
1.6 Seminar Ablauf 15
1.7 Agenda – Einführung in Spark 16
1.8 Spark Überblick 17
1.8.1 Spark Engine 18
1.8.2 Spark Bibliotheken 19
1.8.3 Spark Programmierung 20
1.8.4 Spark Management 21
1.8.5 Spark Storage 22
1.9 Spark im Hadoop Kontext 23
1.10 Grundproblem Big Data 24
1.11 MapReduce 26
1.11.1 MapReduce Programmiermodell 27
1.11.2 Probleme der Hadoop MapReduce Implementierung 28
1.11.3 MapReduce Wordcount Beispiel 29
1.12 Lösungen 30
1.13 Spark Wordcount Beispiel 31
1.14 MapReduce vs. Spark 32
1.15 Spark Kernidee 33
1.16 Architektur: Driver und Executors 34
1.17 Arbeiten im Seminar - Infrastruktur 35

2 Spark Core 40

2.1 Agenda 41
2.2 Einordnung 42
2.3 Resilient Distributed Dataset (RDD) 43
2.4 Konzept 44
2.4.1 Laziness 45
2.4.2 Partitionierung 46
2.4.3 Unveränderlichkeit 47
2.4.4 Lineage 48
2.5 Programmier-Modell in Spark 49
2.6 SparkContext 50
2.7 Erstellen von RDDs 51
2.8 RDD Operationen 52
2.9 Shuffling 53
2.10 Transformations 54
2.11 Actions 55
2.12 Exkurs: Lambda in Python 56
2.13 Transformations (mit Lambda) 57
2.14 Actions (mit Lambda) 58
2.15 Key-Value RDDs 59
2.16 Transformations (Key-Value RDDs) 60
2.17 Actions (Key-Value RDDs) 61
2.18 Best Practice 62
2.18.1 groupByKey vs. reduceByKey 62
2.18.2 reduceByKey vs. aggregateByKey 63
2.18.3 flatMap – join – groupBy 64
2.18.4 Wann lohnt sich Shuffling? 65
2.19 Partitionierung 66
2.20 Persistenz 67
2.21 Storage Level 68
2.22 Closures 69

2.22.1 Closures in Spark ... 70
2.22.2 Beispiele ... 71
2.23 Geteilte Variablen ... 72
2.23.1 Broadcast-Variablen ... 73
2.23.2 Accumulator-Variablen ... 74
2.24 Ablauf ... 75

3 Architektur ... 76

3.1 Agenda ... 77
3.2 Spark Applikation – Übersicht ... 78
3.3 Directed Acyclic Graph (DAG) ... 79
3.4 Spark Architektur ... 81
3.5 Deployment ... 82
3.5.1 Deployment Mode – Local Mode ... 82
3.5.2 Deployment Mode – Client Mode ... 83
3.5.3 Deployment Mode – Cluster Mode ... 84
3.6 Cluster Manager ... 85
3.6.1 Spark Local Mode ... 86
3.6.2 Spark mit YARN – Client Mode ... 87
3.6.3 Spark mit YARN – Cluster Mode ... 88
3.6.4 Spark mit Mesos ... 89
3.6.5 Spark Standalone ... 90
3.7 Pyspark ... 91
3.8 Exkurs in die Java Welt ... 92
3.9 Typen von RDDs ... 93
3.9.1 Typen von RDDs – Beispiele ... 94
3.9.2 Weitere Typen ... 95
3.10 RDD Abstract Class ... 96
3.10.1 Beispiel: HadoopRDD ... 97
3.10.2 Beispiel: FilteredRDD ... 98
3.11 Funktionen (Java) ... 99
3.12 Python vs. Scala vs. Java ... 100
3.13 Ablauf ... 101

4 Spark SQL ... 102

4.1 Ablauf ... 103
4.2 Agenda ... 104
4.3 Einordnung ... 105
4.4 Prozedurale vs. relationale API ... 106
4.5 DataFrame vs. SQL ... 107
4.6 DataFrame (relationale API) ... 108
4.6.1 Caching von DataFrames ... 109
4.6.2 Repräsentation von DataFrames ... 110
4.6.3 Datentypen in Spark SQL ... 111
4.6.4 Datenstruktur in Spark SQL ... 112
4.6.5 Wo finde ich Funktionen? ... 113
4.7 Ausführungspläne ... 114
4.8 Catalyst Optimizer ... 115
4.9 SparkSession ... 116
4.10 Zusammenfassung Spark SQL Architektur ... 117
4.11 Typischer Workflow ... 118
4.11.1 Workflow: Einlesen ... 119
4.11.2 Workflow: Transformieren ... 120
4.12 Typischer Workflow – Einlesen ... 123
4.12.1 Datenquellen ... 124
4.12.2 Daten aus dem HDFS laden ... 125
4.12.3 Daten aus Hive laden ... 126
4.12.4 Daten aus HBase laden ... 127
4.12.5 Python Objekte aus dem Driver laden ... 128
4.12.6 Pandas Dataframe aus dem Driver laden ... 129

4.12.7 Daten aus einer relationalen Datenbank laden ... 130
4.12.8 Daten verifizieren ... 131
4.12.9 Spalten verändern ... 132
4.12.10 Umgang mit Null-Werten ... 133
4.13 Typischer Workflow – Transformieren ... 134
4.13.1 Expressions ... 135
4.13.2 Zeilen filtern ... 136
4.13.3 Spalten ändern und hinzufügen ... 137
4.13.4 Funktionen ... 138
4.13.5 Bedingte Änderungen ... 140
4.13.6 Benutzerdefinierte Funktionen ... 141
4.13.7 Aggregieren ... 142
4.13.8 Joins ... 143
4.14 Typischer Workflow – Exportieren ... 144
4.14.1 Daten speichern ... 145
4.14.2 Daten schreiben – Pattern ... 146
4.14.3 Daten in den Driver laden ... 147
4.15 SQL API ... 148
4.15.1 Tables und Views ... 149
4.15.2 Databases und Catalog ... 150
4.15.3 Tables erstellen ... 151
4.15.4 Views erstellen ... 152
4.15.5 Queries ... 153
4.15.6 Beispiel: Iteratives Vorgehen mit Tables und Views ... 154
4.16 Nahtlose Integration der APIs ... 155
4.17 Fazit Spark SQL ... 156
4.18 Runtime safety: SQL vs. DataFrame vs. Dataset ... 157

5 Administration ... 158

5.1 Agenda ... 159
5.2 Monitoring ... 160
5.3 Testing ... 161
5.4 Spark-Konfiguration ... 162
5.5 Application Properties ... 163
5.6 Spark Properties ... 164
5.7 Memory Management ... 165
5.7.1 OOM Driver ... 166
5.7.2 OOM Executor ... 167
5.8 Security ... 169
5.9 Spark Speculation ... 170
5.10 Spark mit YARN - Ressourcen ... 171
5.11 Spark mit YARN – Settings ... 172
5.12 YARN Scheduler ... 173
5.12.1 FIFO Scheduler ... 174
5.12.2 Capacity Scheduler ... 175
5.12.3 Fair Scheduler ... 176
5.12.4 YARN Scheduler Konfiguration ... 177
5.12.5 YARN Scheduler Web UI ... 178
5.13 „yarn“ Kommando ... 179
5.14 Ablauf ... 180

6 Weitere Bibliotheken ... 181

6.1 Agenda ... 182
6.2 Spark Streaming ... 183
6.2.1 Prinzip ... 184
6.2.2 Streaming Context ... 185
6.2.3 DStreams ... 186
6.2.4 Erstellen von DStreams ... 187
6.2.5 Operationen auf DStreams ... 188
6.2.6 Data Flow ... 189

6.2.7 Spark Structured Streaming ... 190
6.2.8 Datenstrom als Tabelle ... 191
6.2.9 Datenverarbeitung ... 192
6.2.10 Output Modus ... 193
6.3 Machine Learning ... 194
6.3.1 Machine Learning Workflow ... 195
6.3.2 Spark MLlib ... 196
6.3.3 ML-Algorithmen ... 197
6.3.4 Decision Tree ... 198
6.3.5 Lineare Regression ... 199
6.3.6 Logistische Regression ... 200
6.3.7 ML-Datentypen ... 201
6.3.8 Spark MLlib APIs ... 202
6.3.9 MLlib RDD API: Decision Tree ... 203
6.3.10 MLlib RDD API: Lineare Regression ... 204
6.3.11 MLlib DataFrame API ... 205
6.3.12 ML-Pipelines ... 206
6.3.13 PipelineModel ... 207
6.3.14 Beispiele ... 208
6.4 GraphX ... 209
6.4.1 Graphverarbeitung ... 210
6.4.2 Beispiele ... 211
6.4.3 GraphX vs. Graphframes ... 212
6.4.4 Property Graph ... 213
6.4.5 Graph-Operationen ... 214
6.5 Ablauf ... 215

7 Fazit und Ausblick ... 216

7.1 Agenda ... 217
7.2 Zusammenfassung ... 218
7.3 Ausblick und News ... 219
7.4 Literatur ... 220