
Table of Contents

Foreword.....	xix
Preface.....	xxi
1. Secondary Sort: Introduction.....	1
Solutions to the Secondary Sort Problem	3
Implementation Details	3
Data Flow Using Plug-in Classes	6
MapReduce/Hadoop Solution to Secondary Sort	7
Input	7
Expected Output	7
map() Function	8
reduce() Function	8
Hadoop Implementation Classes	9
Sample Run of Hadoop Implementation	10
How to Sort in Ascending or Descending Order	12
Spark Solution to Secondary Sort	12
Time Series as Input	12
Expected Output	13
Option 1: Secondary Sorting in Memory	13
Spark Sample Run	20
Option #2: Secondary Sorting Using the Spark Framework	24
Further Reading on Secondary Sorting	25
2. Secondary Sort: A Detailed Example.....	27
Secondary Sorting Technique	28
Complete Example of Secondary Sorting	32
Input Format	32

Output Format	33
Composite Key	33
Sample Run—Old Hadoop API	36
Input	36
Running the MapReduce Job	37
Output	37
Sample Run—New Hadoop API	37
Input	38
Running the MapReduce Job	38
Output	39
3. Top 10 List.....	41
Top N, Formalized	42
MapReduce/Hadoop Implementation: Unique Keys	43
Implementation Classes in MapReduce/Hadoop	47
Top 10 Sample Run	47
Finding the Top 5	49
Finding the Bottom 10	49
Spark Implementation: Unique Keys	50
RDD Refresher	50
Spark's Function Classes	51
Review of the Top N Pattern for Spark	52
Complete Spark Top 10 Solution	53
Sample Run: Finding the Top 10	58
Parameterizing Top N	59
Finding the Bottom N	61
Spark Implementation: Nonunique Keys	62
Complete Spark Top 10 Solution	64
Sample Run	72
Spark Top 10 Solution Using takeOrdered()	73
Complete Spark Implementation	74
Finding the Bottom N	79
Alternative to Using takeOrdered()	80
MapReduce/Hadoop Top 10 Solution: Nonunique Keys	81
Sample Run	82
4. Left Outer Join.....	85
Left Outer Join Example	85
Example Queries	87
Implementation of Left Outer Join in MapReduce	88
MapReduce Phase 1: Finding Product Locations	88
MapReduce Phase 2: Counting Unique Locations	92

Implementation Classes in Hadoop	93
Sample Run	93
Spark Implementation of Left Outer Join	95
Spark Program	97
Running the Spark Solution	104
Running Spark on YARN	106
Spark Implementation with leftOuterJoin()	107
Spark Program	109
Sample Run on YARN	116
5. Order Inversion.....	119
Example of the Order Inversion Pattern	120
MapReduce/Hadoop Implementation of the Order Inversion Pattern	122
Custom Partitioner	123
Relative Frequency Mapper	124
Relative Frequency Reducer	126
Implementation Classes in Hadoop	127
Sample Run	127
Input	127
Running the MapReduce Job	127
Generated Output	128
6. Moving Average.....	131
Example 1: Time Series Data (Stock Prices)	131
Example 2: Time Series Data (URL Visits)	132
Formal Definition	133
POJO Moving Average Solutions	134
Solution 1: Using a Queue	134
Solution 2: Using an Array	135
Testing the Moving Average	136
Sample Run	136
MapReduce/Hadoop Moving Average Solution	137
Input	137
Output	137
Option #1: Sorting in Memory	138
Sample Run	141
Option #2: Sorting Using the MapReduce Framework	143
Sample Run	147
7. Market Basket Analysis.....	151
MBA Goals	151
Application Areas for MBA	153

Market Basket Analysis Using MapReduce	153
Input	154
Expected Output for Tuple2 (Order of 2)	155
Expected Output for Tuple3 (Order of 3)	155
Informal Mapper	155
Formal Mapper	156
Reducer	157
MapReduce/Hadoop Implementation Classes	158
Sample Run	162
Spark Solution	163
MapReduce Algorithm Workflow	165
Input	166
Spark Implementation	166
YARN Script for Spark	178
Creating Item Sets from Transactions	178
8. Common Friends.....	181
Input	182
POJO Common Friends Solution	182
MapReduce Algorithm	183
The MapReduce Algorithm in Action	184
Solution 1: Hadoop Implementation Using Text	187
Sample Run for Solution 1	187
Solution 2: Hadoop Implementation Using ArrayListOfLongsWritable	189
Sample Run for Solution 2	189
Spark Solution	190
Spark Program	191
Sample Run of Spark Program	197
9. Recommendation Engines Using MapReduce.....	201
Customers Who Bought This Item Also Bought	202
Input	202
Expected Output	202
MapReduce Solution	203
Frequently Bought Together	206
Input and Expected Output	207
MapReduce Solution	208
Recommend Connection	211
Input	213
Output	214
MapReduce Solution	214
Spark Implementation	216

Sample Run of Spark Program	222
10. Content-Based Recommendation: Movies.....	227
Input	228
MapReduce Phase 1	229
MapReduce Phases 2 and 3	229
MapReduce Phase 2: Mapper	230
MapReduce Phase 2: Reducer	231
MapReduce Phase 3: Mapper	233
MapReduce Phase 3: Reducer	234
Similarity Measures	236
Movie Recommendation Implementation in Spark	236
High-Level Solution in Spark	237
Sample Run of Spark Program	250
11. Smarter Email Marketing with the Markov Model.....	257
Markov Chains in a Nutshell	258
Markov Model Using MapReduce	261
Generating Time-Ordered Transactions with MapReduce	262
Hadoop Solution 1: Time-Ordered Transactions	263
Hadoop Solution 2: Time-Ordered Transactions	264
Generating State Sequences	268
Generating a Markov State Transition Matrix with MapReduce	271
Using the Markov Model to Predict the Next Smart Email Marketing Date	274
Spark Solution	275
Input Format	275
High-Level Steps	276
Spark Program	277
Script to Run the Spark Program	286
Sample Run	287
12. K-Means Clustering.....	289
What Is K-Means Clustering?	292
Application Areas for Clustering	292
Informal K-Means Clustering Method: Partitioning Approach	293
K-Means Distance Function	294
K-Means Clustering Formalized	295
MapReduce Solution for K-Means Clustering	295
MapReduce Solution: map()	297
MapReduce Solution: combine()	298
MapReduce Solution: reduce()	299
K-Means Implementation by Spark	300

Sample Run of Spark K-Means Implementation	302
13. k-Nearest Neighbors.....	305
kNN Classification	306
Distance Functions	307
kNN Example	308
An Informal kNN Algorithm	308
Formal kNN Algorithm	309
Java-like Non-MapReduce Solution for kNN	309
kNN Implementation in Spark	311
Formalizing kNN for the Spark Implementation	312
Input Data Set Formats	313
Spark Implementation	313
YARN shell script	325
14. Naive Bayes.....	327
Training and Learning Examples	328
Numeric Training Data	328
Symbolic Training Data	329
Conditional Probability	331
The Naive Bayes Classifier in Depth	331
Naive Bayes Classifier Example	332
The Naive Bayes Classifier: MapReduce Solution for Symbolic Data	334
Stage 1: Building a Classifier Using Symbolic Training Data	335
Stage 2: Using the Classifier to Classify New Symbolic Data	341
The Naive Bayes Classifier: MapReduce Solution for Numeric Data	343
Naive Bayes Classifier Implementation in Spark	345
Stage 1: Building a Classifier Using Training Data	346
Stage 2: Using the Classifier to Classify New Data	355
Using Spark and Mahout	361
Apache Spark	361
Apache Mahout	362
15. Sentiment Analysis.....	363
Sentiment Examples	364
Sentiment Scores: Positive or Negative	364
A Simple MapReduce Sentiment Analysis Example	365
map() Function for Sentiment Analysis	366
reduce() Function for Sentiment Analysis	367
Sentiment Analysis in the Real World	367

16. Finding, Counting, and Listing All Triangles in Large Graphs.....	369
Basic Graph Concepts	370
Importance of Counting Triangles	372
MapReduce/Hadoop Solution	372
Step 1: MapReduce in Action	373
Step 2: Identify Triangles	375
Step 3: Remove Duplicate Triangles	376
Hadoop Implementation Classes	377
Sample Run	377
Spark Solution	380
High-Level Steps	380
Sample Run	387
17. K-mer Counting.....	391
Input Data for K-mer Counting	392
Sample Data for K-mer Counting	392
Applications of K-mer Counting	392
K-mer Counting Solution in MapReduce/Hadoop	393
The map() Function	393
The reduce() Function	394
Hadoop Implementation Classes	394
K-mer Counting Solution in Spark	395
Spark Solution	396
Sample Run	405
18. DNA Sequencing.....	407
Input Data for DNA Sequencing	409
Input Data Validation	410
DNA Sequence Alignment	411
MapReduce Algorithms for DNA Sequencing	412
Step 1: Alignment	415
Step 2: Recalibration	423
Step 3: Variant Detection	428
19. Cox Regression.....	433
The Cox Model in a Nutshell	434
Cox Regression Basic Terminology	435
Cox Regression Using R	436
Expression Data	436
Cox Regression Application	437
Cox Regression POJO Solution	437
Input for MapReduce	439

Input Format	440
Cox Regression Using MapReduce	440
Cox Regression Phase 1: map()	440
Cox Regression Phase 1: reduce()	441
Cox Regression Phase 2: map()	442
Sample Output Generated by Phase 1 reduce() Function	444
Sample Output Generated by the Phase 2 map() Function	445
Cox Regression Script for MapReduce	445
20. Cochran-Armitage Test for Trend.....	447
Cochran-Armitage Algorithm	448
Application of Cochran-Armitage	454
MapReduce Solution	456
Input	456
Expected Output	457
Mapper	458
Reducer	459
MapReduce/Hadoop Implementation Classes	463
Sample Run	463
21. Allelic Frequency.....	465
Basic Definitions	466
Chromosome	466
Bioset	467
Allele and Allelic Frequency	467
Source of Data for Allelic Frequency	467
Allelic Frequency Analysis Using Fisher's Exact Test	469
Fisher's Exact Test	469
Formal Problem Statement	471
MapReduce Solution for Allelic Frequency	471
MapReduce Solution, Phase 1	472
Input	472
Output/Result	473
Phase 1 Mapper	474
Phase 1 Reducer	475
Sample Run of Phase 1 MapReduce/Hadoop Implementation	479
Sample Plot of P-Values	481
MapReduce Solution, Phase 2	482
Phase 2 Mapper for Bottom 100 P-Values	482
Phase 2 Reducer for Bottom 100 P-Values	484
Is Our Bottom 100 List a Monoid?	485
Hadoop Implementation Classes for Bottom 100 List	486

MapReduce Solution, Phase 3	486
Phase 3 Mapper for Bottom 100 P-Values	487
Phase 3 Reducer for Bottom 100 P-Values	489
Hadoop Implementation Classes for Bottom 100 List for Each Chromosome	490
Special Handling of Chromosomes X and Y	490
22. The T-Test.....	491
Performing the T-Test on Biosets	492
MapReduce Problem Statement	495
Input	496
Expected Output	496
MapReduce Solution	496
Hadoop Implementation Classes	499
Spark Implementation	499
High-Level Steps	500
T-Test Algorithm	507
Sample Run	509
23. Pearson Correlation.....	513
Pearson Correlation Formula	514
Pearson Correlation Example	516
Data Set for Pearson Correlation	517
POJO Solution for Pearson Correlation	517
POJO Solution Test Drive	518
MapReduce Solution for Pearson Correlation	519
map() Function for Pearson Correlation	519
reduce() Function for Pearson Correlation	520
Hadoop Implementation Classes	521
Spark Solution for Pearson Correlation	522
Input	523
Output	523
Spark Solution	524
High-Level Steps	525
Step 1: Import required classes and interfaces	527
smaller() method	528
MutableDouble class	529
toMap() method	530
toListOfString() method	530
readBiosets() method	531
Step 2: Handle input parameters	532
Step 3: Create a Spark context object	533

Step 4: Create list of input files/biomarkers	534
Step 5: Broadcast reference as global shared object	534
Step 6: Read all biomarkers from HDFS and create the first RDD	534
Step 7: Filter biomarkers by reference	535
Step 8: Create (Gene-ID, (Patient-ID, Gene-Value)) pairs	536
Step 9: Group by gene	537
Step 10: Create Cartesian product of all genes	538
Step 11: Filter redundant pairs of genes	538
Step 12: Calculate Pearson correlation and p-value	539
Pearson Correlation Wrapper Class	542
Testing the Pearson Class	543
Pearson Correlation Using R	543
YARN Script to Run Spark Program	544
Spearman Correlation Using Spark	544
Spearman Correlation Wrapper Class	544
Testing the Spearman Correlation Wrapper Class	545
24. DNA Base Count.....	547
FASTA Format	548
FASTA Format Example	549
FASTQ Format	549
FASTQ Format Example	549
MapReduce Solution: FASTA Format	550
Reading FASTA Files	550
MapReduce FASTA Solution: map()	550
MapReduce FASTA Solution: reduce()	551
Sample Run	552
Log of sample run	552
Generated output	552
Custom Sorting	553
Custom Partitioning	554
MapReduce Solution: FASTQ Format	556
Reading FASTQ Files	557
MapReduce FASTQ Solution: map()	558
MapReduce FASTQ Solution: reduce()	559
Hadoop Implementation Classes: FASTQ Format	560
Sample Run	560
Spark Solution: FASTA Format	561
High-Level Steps	561
Sample Run	561
Spark Solution: FASTQ Format	564
High-Level Steps	566
	566

Step 1: Import required classes and interfaces	567
Step 2: Handle input parameters	567
Step 3: Create a JavaPairRDD from FASTQ input	568
Step 4: Map partitions	568
Step 5: Collect all DNA base counts	569
Step 6: Emit Final Counts	570
Sample Run	570
25. RNA Sequencing.....	573
Data Size and Format	574
MapReduce Workflow	574
Input Data Validation	574
RNA Sequencing Analysis Overview	575
MapReduce Algorithms for RNA Sequencing	578
Step 1: MapReduce TopHat Mapping	579
Step 2: MapReduce Calling Cuffdiff	582
26. Gene Aggregation.....	585
Input	586
Output	586
MapReduce Solutions (Filter by Individual and by Average)	587
Mapper: Filter by Individual	588
Reducer: Filter by Individual	590
Mapper: Filter by Average	590
Reducer: Filter by Average	592
Computing Gene Aggregation	592
Hadoop Implementation Classes	594
Analysis of Output	597
Gene Aggregation in Spark	600
Spark Solution: Filter by Individual	601
Sharing Data Between Cluster Nodes	601
High-Level Steps	602
Utility Functions	607
Sample Run	609
Spark Solution: Filter by Average	610
High-Level Steps	611
Utility Functions	616
Sample Run	619
27. Linear Regression.....	621
Basic Definitions	622
Simple Example	622

Problem Statement	624
Input Data	625
Expected Output	625
MapReduce Solution Using SimpleRegression	626
Hadoop Implementation Classes	628
MapReduce Solution Using R's Linear Model	629
Phase 1	630
Phase 2	633
Hadoop Implementation Using Classes	635
28. MapReduce and Monoids.....	637
Introduction	637
Definition of Monoid	639
How to Form a Monoid	640
Monoidic and Non-Monoidic Examples	640
Maximum over a Set of Integers	641
Subtraction over a Set of Integers	641
Addition over a Set of Integers	641
Multiplication over a Set of Integers	641
Mean over a Set of Integers	642
Non-Commutative Example	642
Median over a Set of Integers	642
Concatenation over Lists	642
Union/Intersection over Integers	643
Functional Example	643
Matrix Example	644
MapReduce Example: Not a Monoid	644
MapReduce Example: Monoid	646
Hadoop Implementation Classes	647
Sample Run	648
View Hadoop output	650
Spark Example Using Monoids	650
High-Level Steps	652
Sample Run	656
Conclusion on Using Monoids	657
Functors and Monoids	658
29. The Small Files Problem.....	661
Solution 1: Merging Small Files Client-Side	662
Input Data	665
Solution with SmallFilesConsolidator	665
Solution Without SmallFilesConsolidator	667

Solution 2: Solving the Small Files Problem with CombineFileInputFormat	668
Custom CombineFileInputFormat	672
Sample Run Using CustomCFIF	672
Alternative Solutions	674
30. Huge Cache for MapReduce.....	675
Implementation Options	676
Formalizing the Cache Problem	677
An Elegant, Scalable Solution	678
Implementing the LRUMap Cache	681
Extending the LRUMap Class	681
Testing the Custom Class	682
The MapDBEntry Class	683
Using MapDB	684
Testing MapDB: put()	686
Testing MapDB: get()	687
MapReduce Using the LRUMap Cache	687
CacheManager Definition	688
Initializing the Cache	689
Using the Cache	690
Closing the Cache	691
31. The Bloom Filter.....	693
Bloom Filter Properties	693
A Simple Bloom Filter Example	696
Bloom Filters in Guava Library	696
Using Bloom Filters in MapReduce	698
A. Bioset.....	699
B. Spark RDDs.....	701
Bibliography.....	721
Index.....	725