

---

# Hadoop Application Architectures

*Mark Grover, Ted Malaska,  
Jonathan Seidman & Gwen Shapira*

Beijing • Boston • Farnham • Sebastopol • Tokyo

**O'REILLY**

---

# Table of Contents

**Foreword**..... ix

**Preface**..... xi

---

## **Part I. Architectural Considerations for Hadoop Applications**

<b>1. Data Modeling in Hadoop</b> .....	<b>1</b>
Data Storage Options	2
Standard File Formats	4
Hadoop File Types	5
Serialization Formats	7
Columnar Formats	9
Compression	12
HDFS Schema Design	14
Location of HDFS Files	16
Advanced HDFS Schema Design	17
HDFS Schema Design Summary	21
HBase Schema Design	21
Row Key	22
Timestamp	25
Hops	25
Tables and Regions	26
Using Columns	28
Using Column Families	30
Time-to-Live	30
Managing Metadata	31
What Is Metadata?	31
Why Care About Metadata?	32
Where to Store Metadata?	32

---

Examples of Managing Metadata	34
Limitations of the Hive Metastore and HCatalog	34
Other Ways of Storing Metadata	35
Conclusion	36
<b>2. Data Movement.....</b>	<b>39</b>
Data Ingestion Considerations	39
Timeliness of Data Ingestion	40
Incremental Updates	42
Access Patterns	43
Original Source System and Data Structure	44
Transformations	47
Network Bottlenecks	48
Network Security	49
Push or Pull	49
Failure Handling	50
Level of Complexity	51
Data Ingestion Options	51
File Transfers	52
Considerations for File Transfers versus Other Ingest Methods	55
Sqoop: Batch Transfer Between Hadoop and Relational Databases	56
Flume: Event-Based Data Collection and Processing	61
Kafka	71
Data Extraction	76
Conclusion	77
<b>3. Processing Data in Hadoop.....</b>	<b>79</b>
MapReduce	80
MapReduce Overview	80
Example for MapReduce	88
When to Use MapReduce	94
Spark	95
Spark Overview	95
Overview of Spark Components	96
Basic Spark Concepts	97
Benefits of Using Spark	100
Spark Example	102
When to Use Spark	104
Abstractions	104
Pig	106
Pig Example	106
When to Use Pig	109

Crunch	110
Crunch Example	110
When to Use Crunch	115
Cascading	115
Cascading Example	116
When to Use Cascading	119
Hive	119
Hive Overview	119
Example of Hive Code	121
When to Use Hive	125
Impala	126
Impala Overview	127
Speed-Oriented Design	128
Impala Example	130
When to Use Impala	131
Conclusion	132
<b>4. Common Hadoop Processing Patterns. ....</b>	<b>135</b>
Pattern: Removing Duplicate Records by Primary Key	135
Data Generation for Deduplication Example	136
Code Example: Spark Deduplication in Scala	137
Code Example: Deduplication in SQL	139
Pattern: Windowing Analysis	140
Data Generation for Windowing Analysis Example	141
Code Example: Peaks and Valleys in Spark	142
Code Example: Peaks and Valleys in SQL	146
Pattern: Time Series Modifications	147
Use HBase and Versioning	148
Use HBase with a RowKey of RecordKey and StartTime	149
Use HDFS and Rewrite the Whole Table	149
Use Partitions on HDFS for Current and Historical Records	150
Data Generation for Time Series Example	150
Code Example: Time Series in Spark	151
Code Example: Time Series in SQL	154
Conclusion	157
<b>5. Graph Processing on Hadoop. ....</b>	<b>159</b>
What Is a Graph?	159
What Is Graph Processing?	161
How Do You Process a Graph in a Distributed System?	162
The Bulk Synchronous Parallel Model	163
BSP by Example	163

Giraph	165
Read and Partition the Data	166
Batch Process the Graph with BSP	168
Write the Graph Back to Disk	172
Putting It All Together	173
When Should You Use Giraph?	174
GraphX	174
Just Another RDD	175
GraphX Pregel Interface	177
vprog()	178
sendMessage()	179
mergeMessage()	179
Which Tool to Use?	180
Conclusion	180
<b>6. Orchestration.....</b>	<b>183</b>
Why We Need Workflow Orchestration	183
The Limits of Scripting	184
The Enterprise Job Scheduler and Hadoop	186
Orchestration Frameworks in the Hadoop Ecosystem	186
Oozie Terminology	188
Oozie Overview	188
Oozie Workflow	191
Workflow Patterns	194
Point-to-Point Workflow	194
Fan-Out Workflow	196
Capture-and-Decide Workflow	198
Parameterizing Workflows	201
Classpath Definition	203
Scheduling Patterns	204
Frequency Scheduling	205
Time and Data Triggers	205
Executing Workflows	210
Conclusion	210
<b>7. Near-Real-Time Processing with Hadoop.....</b>	<b>213</b>
Stream Processing	215
Apache Storm	217
Storm High-Level Architecture	218
Storm Topologies	219
Tuples and Streams	221
Spouts and Bolts	221

Stream Groupings	222
Reliability of Storm Applications	223
Exactly-Once Processing	223
Fault Tolerance	224
Integrating Storm with HDFS	225
Integrating Storm with HBase	225
Storm Example: Simple Moving Average	226
Evaluating Storm	233
Trident	233
Trident Example: Simple Moving Average	234
Evaluating Trident	237
Spark Streaming	237
Overview of Spark Streaming	238
Spark Streaming Example: Simple Count	238
Spark Streaming Example: Multiple Inputs	240
Spark Streaming Example: Maintaining State	241
Spark Streaming Example: Windowing	243
Spark Streaming Example: Streaming versus ETL Code	244
Evaluating Spark Streaming	245
Flume Interceptors	246
Which Tool to Use?	247
Low-Latency Enrichment, Validation, Alerting, and Ingestion	247
NRT Counting, Rolling Averages, and Iterative Processing	248
Complex Data Pipelines	249
Conclusion	250

---

## Part II. Case Studies

<b>8. Clickstream Analysis.....</b>	<b>253</b>
Defining the Use Case	253
Using Hadoop for Clickstream Analysis	255
Design Overview	256
Storage	257
Ingestion	260
The Client Tier	264
The Collector Tier	266
Processing	268
Data Deduplication	270
Sessionization	272
Analyzing	275
Orchestration	276

Conclusion	279
<b>9. Fraud Detection</b>	<b>281</b>
Continuous Improvement	281
Taking Action	282
Architectural Requirements of Fraud Detection Systems	283
Introducing Our Use Case	283
High-Level Design	284
Client Architecture	286
Profile Storage and Retrieval	287
Caching	288
HBase Data Definition	289
Delivering Transaction Status: Approved or Denied?	294
Ingest	295
Path Between the Client and Flume	296
Near-Real-Time and Exploratory Analytics	302
Near-Real-Time Processing	302
Exploratory Analytics	304
What About Other Architectures?	305
Flume Interceptors	305
Kafka to Storm or Spark Streaming	306
External Business Rules Engine	306
Conclusion	307
<b>10. Data Warehouse</b>	<b>309</b>
Using Hadoop for Data Warehousing	312
Defining the Use Case	314
OLTP Schema	316
Data Warehouse: Introduction and Terminology	317
Data Warehousing with Hadoop	319
High-Level Design	319
Data Modeling and Storage	320
Ingestion	332
Data Processing and Access	337
Aggregations	341
Data Export	343
Orchestration	344
Conclusion	345
<b>A. Joins in Impala</b>	<b>347</b>
<b>Index</b>	<b>353</b>