Making Everything Easier!™

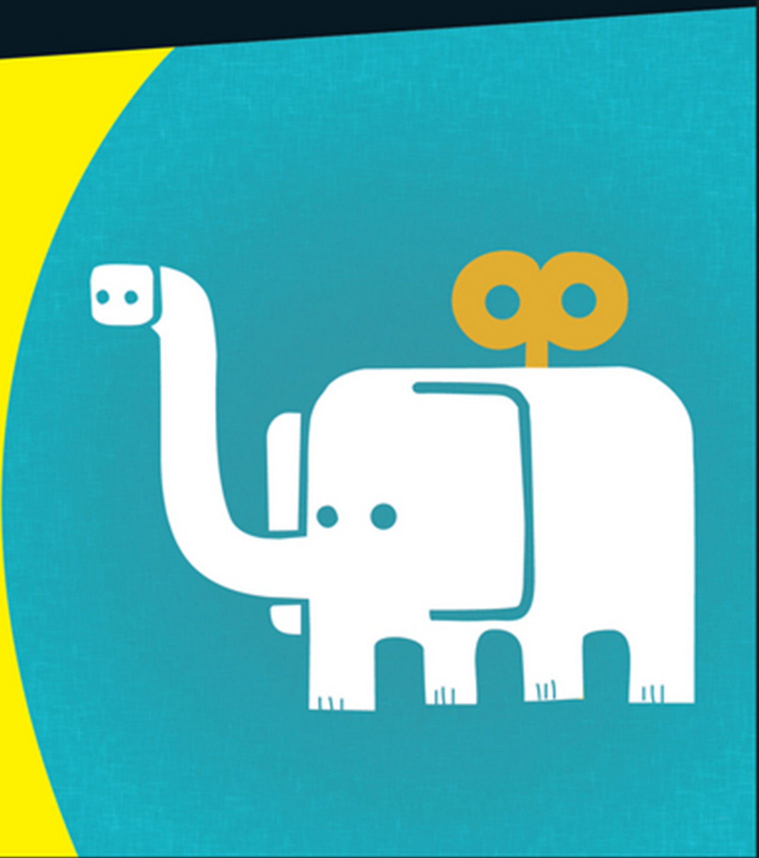
Hadoop®

FOR DUMMIES®

A Wiley Brand

Learn to:

- Understand the value of big data and how Hadoop can help manage it
- Navigate the Hadoop 2 ecosystem and create clusters
- Use applications for data mining, problem-solving, analytics, and more



Dirk deRoos
Paul C. Zikopoulos
Roman B. Melnyk, PhD
Bruce Brown
Rafael Coss

Hadoop® For Dummies®

A Wiley Brand

by Dirk deRoos, Paul C. Zikopoulos, Bruce Brown, Rafael Coss, and Roman B. Melnyk

For Dummies®
A Wiley Brand

Hadoop® For Dummies®

Published by: **John Wiley & Sons, Inc.,** 111 River Street, Hoboken, NJ 07030-5774, www.wiley.com

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002. For technical support, please visit www.wiley.com/techsupport.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2013954209

ISBN: 978-1-118-60755-8 (pbk); ISBN 978-1-118-65220-6 (ebk); ISBN 978-1-118-70503-2 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Contents at a Glance

Introduction *1*

Part I: Getting Started with Hadoop *7*

Chapter 1: Introducing Hadoop and Seeing What It's Good For..................................9

Chapter 2: Common Use Cases for Big Data in Hadoop...23

Chapter 3: Setting Up Your Hadoop Environment...41

Part II: How Hadoop Works *51*

Chapter 4: Storing Data in Hadoop: The Hadoop Distributed File System...............53

Chapter 5: Reading and Writing Data...69

Chapter 6: MapReduce Programming...83

Chapter 7: Frameworks for Processing Data in Hadoop: YARN and MapReduce...103

Chapter 8: Pig: Hadoop Programming Made Easier 117

Chapter 9: Statistical Analysis in Hadoop...129

Chapter 10: Developing and Scheduling Application Workflows with Oozie.........139

Part III: Hadoop and Structured Data *155*

Chapter 11: Hadoop and the Data Warehouse: Friends or Foes?............................157

Chapter 12: Extremely Big Tables: Storing Data in HBase.......................................179

Chapter 13: Applying Structure to Hadoop Data with Hive.....................................227

Chapter 14: Integrating Hadoop with Relational Databases Using Sqoop..............269

Chapter 15: The Holy Grail: Native SQL Access to Hadoop Data 303

Part IV: Administering and Configuring Hadoop *313*

Chapter 16: Deploying Hadoop 315

Chapter 17: Administering Your Hadoop Cluster...335

Part V: The Part of Tens *359*

Chapter 18: Ten Hadoop Resources Worthy of a Bookmark 361

Chapter 19: Ten Reasons to Adopt Hadoop..371

Index *379*

Table of Contents

Introduction 1

About this Book 1
Foolish Assumptions 2
How This Book Is Organized 2
Part I: Getting Started With Hadoop 2
Part II: How Hadoop Works 2
Part III: Hadoop and Structured Data 3
Part IV: Administering and Configuring Hadoop 3
Part V: The Part Of Tens: Getting More Out of Your Hadoop Cluster 3
Icons Used in This Book 3
Beyond the Book 4
Where to Go from Here 5

Part I:Getting Started with Hadoop 7

Chapter 1: Introducing Hadoop and Seeing What It's Good For 9

Big Data and the Need for Hadoop 10
Exploding data volumes 11
Varying data structures 12
A playground for data scientists 12
The Origin and Design of Hadoop 13
Distributed processing with MapReduce 13
Apache Hadoop ecosystem 15
Examining the Various Hadoop Offerings 17
Comparing distributions 18
Working with in-database MapReduce 21
Looking at the Hadoop toolbox 21

Chapter 2: Common Use Cases for Big Data in Hadoop 23

The Keys to Successfully Adopting Hadoop (Or, "Please, Can We Keep Him?") 23
Log Data Analysis 24
Data Warehouse Modernization 27
Fraud Detection 29
Risk Modeling 31
Social Sentiment Analysis 32
Image Classification 36
Graph Analysis 38
To Infinity and Beyond 39

Chapter 3: Setting Up Your Hadoop Environment 41
Choosing a Hadoop Distribution 41
Choosing a Hadoop Cluster Architecture 44
Pseudo-distributed mode (single node) 44
Fully distributed mode(a cluster of nodes) 44
The Hadoop For Dummies Environment 44
The Hadoop For Dummies distribution: Apache Bigtop 45
Setting up the Hadoop For Dummies environment 46
The Hadoop For Dummies Sample Data Set: Airline on-time performance 48
Your First Hadoop Program: Hello Hadoop! 49

Part II: How Hadoop Works 51

Chapter 4: Storing Data in Hadoop: The Hadoop Distributed File System 53
Data Storage in HDFS 54
Taking a closer look at data blocks 55
Replicating data blocks 56
Slave node and disk failures 57
Sketching Out the HDFS Architecture 57
Looking at slave nodes 58
Keeping track of data blocks with NameNode 60
Checkpointing updates 63
HDFS Federation 65
HDFS High Availability 66

Chapter 5: Reading and Writing Data 69
Compressing Data 69
Managing Files with the Hadoop File System Commands 72
Ingesting Log Data with Flume 80

Chapter 6: MapReduce Programming. 83
Thinking in Parallel 83
Seeing the Importance of MapReduce 85
Doing Things in Parallel: Breaking Big Problems into Many Bite-Size Pieces 86
Looking at MapReduce application flow 87
Understanding input splits 87
Seeing how key/value pairs fit into the MapReduce application flow 89
Writing MapReduce Applications 94

Getting Your Feet Wet: Writing a Simple MapReduce Application 96
The FlightsByCarrier driver application 96
The FlightsByCarrier mapper 98
The FlightsByCarrier reducer 99
Running the FlightsByCarrier application 100

Chapter 7: Frameworks for Processing Data in Hadoop: YARN and MapReduce 103

Running Applications Before Hadoop 2 104
Tracking JobTracker 105
Tracking TaskTracker 105
Launching a MapReduce application 106
Seeing a World beyond MapReduce 107
Scouting out the YARN architecture 108
Launching a YARN-based application 111
Real-Time and Streaming Applications 113

Chapter 8: Pig: Hadoop Programming Made Easier 115

Admiring the Pig Architecture 116
Going with the Pig Latin Application Flow 117
Working through the ABCs of Pig Latin 119
Uncovering Pig Latin structures 120
Looking at Pig data types and syntax 121
Evaluating Local and Distributed Modes of Running Pig scripts 125
Checking Out the Pig Script Interfaces 126
Scripting with Pig Latin 127

Chapter 9: Statistical Analysis in Hadoop 129

Pumping Up Your Statistical Analysis 129
The limitations of sampling 130
Factors that increase the scale of statistical analysis 130
Running statistical models in MapReduce 131
Machine Learning with Mahout 131
Collaborative filtering 133
Clustering 133
Classifications 134
R on Hadoop 135
The R language 135
Hadoop Integration with R 136

Chapter 10: Developing and Scheduling Application Workflows with Oozie 139

Getting Oozie in Place 140
Developing and Running an Oozie Workflow 142
Writing Oozie workflow definitions 143
Configuring Oozie workflows 151
Running Oozie workflows 151

Scheduling and Coordinating Oozie Workflows 152
Time-based scheduling for Oozie coordinator jobs 152
Time and data availability-based scheduling for Oozie coordinator jobs 153
Running Oozie coordinator jobs 154

Part III: Hadoop and Structured Data 155

Chapter 11: Hadoop and the Data Warehouse: Friends or Foes? . . . 157

Comparing and Contrasting Hadoop with Relational Databases 158
NoSQL data stores 159
ACID versus BASE data stores 160
Structured data storage and processing in Hadoop 163
Modernizing the Warehouse with Hadoop 166
The landing zone 166
A queryable archive of cold warehouse data 168
Hadoop as a data preprocessing engine 172
Data discovery and sandboxes 175

Chapter 12: Extremely Big Tables: Storing Data in HBase 179

Say Hello to HBase 180
Sparse 180
It’s distributed and persistent 181
It has a multidimensional sorted map 182
Understanding the HBase Data Model 182
Understanding the HBase Architecture 186
RegionServers 187
MasterServer 190
Zookeeper and HBase reliability 192
Taking HBase for a Test Run 195
Creating a table 199
Working with Zookeeper 202
Getting Things Done with HBase 203
Working with an HBase Java API client example 206
HBase and the RDBMS world 209
Knowing when HBase makes sense for you? 210
ACID Properties in HBase 210
Transitioning from an RDBMS model to HBase 211
Deploying and Tuning HBase 214
Hardware requirements 215
Deployment Considerations 217
Tuning prerequisites 218
Understanding your data access patterns 220
Pre-Splitting your regions 222
The importance of row key design 223
Tuning major compactions 225

Chapter 13: Applying Structure to Hadoop Data with Hive 227

Saying Hello to Hive 228
Seeing How the Hive is Put Together 229
Getting Started with Apache Hive 231
Examining the Hive Clients 234
- The Hive CLI client 234
- The web browser as Hive client 236
- SQuirreL as Hive client with the JDBC Driver 238

Working with Hive Data Types 240
Creating and Managing Databases and Tables 242
- Managing Hive databases 243
- Creating and managing tables with Hive 244

Seeing How the Hive Data Manipulation Language Works 251
- LOAD DATA examples 251
- INSERT examples 255
- Create Table As Select (CTAS) examples 258

Querying and Analyzing Data 259
- Joining tables with Hive 260
- Improving your Hive queries with indexes 262
- Windowing in HiveQL 264
- Other key HiveQL features 267

Chapter 14: Integrating Hadoop with Relational Databases Using Sqoop 269

The Principles of Sqoop Design 270
Scooping Up Data with Sqoop 271
- Connectors and Drivers 271
- Importing Data with Sqoop 272
- Importing data into HDFS 273
- Importing data into Hive 280
- Importing data into HBase 281
- Importing incrementally 285
- Benefiting from additional Sqoop import features 288

Sending Data Elsewhere with Sqoop 290
- Exporting data from HDFS 291
- Sqoop exports using the Insert approach 293
- Sqoop exports using the Update and Update Insert approach ... 295
- Sqoop exports using call stored procedures 295
- Sqoop exports and transactions 296

Looking at Your Sqoop Input and Output Formatting Options 296
- Getting down to brass tacks: An example of output line-formatting and input-parsing 298

Sqoop 2.0 Preview 301

Chapter 15: The Holy Grail: Native SQL Access to Hadoop Data . . . 303
SQL’s Importance for Hadoop....303
Looking at What SQL Access Actually Means....304
SQL Access and Apache Hive....305
Solutions Inspired by Google Dremel....307
Apache Drill....308
Cloudera Impala....309
IBM Big SQL....309
Pivotal HAWQ....311
Hadapt....311
The SQL Access Big Picture....312

Part IV: Administering and Configuring Hadoop.......... 313

Chapter 16: Deploying Hadoop 315
Working with Hadoop Cluster Components....315
Rack considerations....316
Master nodes....318
Slave nodes....321
Edge nodes....326
Networking....328
Hadoop Cluster Configurations....328
Small....329
Medium....329
Large....330
Alternate Deployment Form Factors....331
Virtualized servers....331
Cloud deployments....332
Sizing Your Hadoop Cluster....332

Chapter 17: Administering Your Hadoop Cluster. 335
Achieving Balance: A Big Factor in Cluster Health....335
Mastering the Hadoop Administration Commands....336
Understanding Factors for Performance....342
Hardware....342
MapReduce....342
Benchmarking....343
Tolerating Faults and Data Reliability....344
Putting Apache Hadoop’s Capacity Scheduler to Good Use....346
Setting Security: The Kerberos Protocol....348
Expanding Your Toolset Options....349
Hue....349
Ambari....351
Hadoop User Experience (Hue)....352
The Hadoop shell....354
Basic Hadoop Configuration Details....355

Part V: The Part of Tens 359

Chapter 18: Ten Hadoop Resources Worthy of a Bookmark 361

Central Nervous System: Apache.org 362
Tweet This 362
Hortonworks University 363
Cloudera University 363
BigDataUniversity.com 365
planet Big Data Blog Aggregator 366
Quora's Apache Hadoop Forum 367
The IBM Big Data Hub 367
Conferences Not to Be Missed 367
The Google Papers That Started It All 368
The Bonus Resource: What Did We Ever Do B.G.? 369

Chapter 19: Ten Reasons to Adopt Hadoop. 371

Hadoop Is Relatively Inexpensive 372
Hadoop Has an Active Open Source Community 373
Hadoop Is Being Widely Adopted in Every Industry 374
Hadoop Can Easily Scale Out As Your Data Grows 374
Traditional Tools Are Integrating with Hadoop 375
Hadoop Can Store Data in Any Format 375
Hadoop Is Designed to Run Complex Analytics 376
Hadoop Can Process a Full Data Set (As Opposed to Sampling) 376
Hardware Is Being Optimized for Hadoop 376
Hadoop Can Increasingly Handle Flexible Workloads
(No Longer Just Batch) 377

Index 379

Introduction

Welcome to *Hadoop for Dummies*! Hadoop is an exciting technology, and this book will help you cut through the hype and wrap your head around what it's good for and how it works. We've included examples and plenty of practical advice so you can get started with your own Hadoop cluster.

About this Book

In our own Hadoop learning activities, we're constantly struck by how little beginner-level content is available. For almost any topic, we see two things: high-level marketing blurbs with pretty pictures; and dense, low-level, narrowly focused descriptions. What are missing are solid entry-level explanations that add substance to the marketing fluff and help someone with little or no background knowledge bridge the gap to the more advanced material. Every chapter in this book was written with this goal in mind: to clearly explain the chapter's concept, explain why it's significant in the Hadoop universe, and show how you can get started with it.

No matter how much (or how little) you know about Hadoop, getting started with the technology is not exactly easy for a number of reasons. In addition to the lack of entry-level content, the rapid pace of change in the Hadoop ecosystem makes it difficult to keep on top of standards. We find that most discussions on Hadoop either cover the older interfaces, and are never updated; or they cover the newer interfaces with little insight into how to bridge the gap from the old technology. In this book, we've taken care to describe the current interfaces, but we also discuss previous standards, which are still commonly used in environments where some of the older interfaces are entrenched.

Here are a few things to keep in mind as you read this book:

- Bold text means that you're meant to type the text just as it appears in the book. The exception is when you're working through a steps list: Because each step is bold, the text to type is not bold.
- Web addresses and programming code appear in monofont. If you're reading a digital version of this book on a device connected to the Internet, note that you can click the web address to visit that website, like this: `www.dummies.com`

Foolish Assumptions

We've written this book so that anyone with a basic understanding of computers and IT can learn about Hadoop. But that said, some experience with databases, programming, and working with Linux would be helpful.

There are some parts of this book that require deeper skills, like the Java coverage in Chapter 6 on MapReduce; but if you haven't programmed in Java before, don't worry. The explanations of how MapReduce works don't require you to be a Java programmer. The Java code is there for people who'll want to try writing their own MapReduce applications. In Part 3, a database background would certainly help you understand the significance of the various Hadoop components you can use to integrate with existing databases and work with relational data. But again, we've written in a lot of background to help provide context for the Hadoop concepts we're describing.

How This Book Is Organized

This book is composed of five parts, with each part telling a major chunk of the Hadoop story. Every part and every chapter was written to be a self-contained unit, so you can pick and choose whatever you want to concentrate on. Because many Hadoop concepts are intertwined, we've taken care to refer to whatever background concepts you might need so you can catch up from other chapters, if needed. To give you an idea of the book's layout, here are the parts of the book and what they're about:

Part I: Getting Started With Hadoop

As the beginning of the book, this part gives a rundown of Hadoop and its ecosystem and the most common ways Hadoop's being used. We also show you how you can set up your own Hadoop environment and run the example code we've included in this book.

Part II: How Hadoop Works

This is the meat of the book, with lots of coverage designed to help you understand the nuts and bolts of Hadoop. We explain the storage and processing architecture, and also how you can write your own applications.

Part III: Hadoop and Structured Data

How Hadoop deals with structured data is arguably the most important debate happening in the Hadoop community today. There are many competing SQL-on-Hadoop technologies, which we survey, but we also take a deep look at the more established Hadoop community projects dedicated to structured data: HBase, Hive, and Sqoop.

Part IV: Administering and Configuring Hadoop

When you're ready to get down to brass tacks and deploy a cluster, this part is a great starting point. Hadoop clusters sink or swim depending on how they're configured and deployed, and we've got loads of experience-based advice here.

Part V: The Part Of Tens: Getting More Out of Your Hadoop Cluster

To cap off the book, we've given you a list of additional places where you can bone up on your Hadoop skills. We've also provided you an additional set of reasons to adopt Hadoop, just in case you weren't convinced already.

Icons Used in This Book

TIP

The Tip icon marks tips (duh!) and shortcuts that you can use to make working with Hadoop easier.

REMEMBER

Remember icons mark the information that's especially important to know. To siphon off the most important information in each chapter, just skim through these icons.

The Technical Stuff icon marks information of a highly technical nature that you can normally skip over.

The Warning icon tells you to watch out! It marks important information that may save you headaches.

Beyond the Book

We have written a lot of extra content that you won't find in this book. Go online to find the following:

- **The Cheat Sheet for this book is at**

  ```
  www.dummies.com/cheatsheet/hadoop
  ```

  Here you'll find quick references for useful Hadoop information we've brought together and keep up to date. For instance, a handy list of the most common Hadoop commands and their syntax, a map of the various Hadoop ecosystem components, and what they're good for, and listings of the various Hadoop distributions available in the market and their unique offerings. Since the Hadoop ecosystem is continually evolving, we've also got instructions on how to set up the *Hadoop for Dummies* environment with the newest production-ready versions of the Hadoop and its components.

- **Updates to this book, if we have any, are at**

  ```
  www.dummies.com/extras/hadoop
  ```

- **Code samples used in this book are also at**

  ```
  www.dummies.com/extras/hadoop
  ```

  All the code samples in this book are posted to the website in Zip format; just download and unzip them and they're ready to use with the *Hadoop for Dummies* environment described in Chapter 3. The Zip files, which are named according to chapter, contain one or more files. Some files have application code (Java, Pig, and Hive) and others have series of commands or scripts. (Refer to the downloadable Read Me file for a detailed description of the files.) Note that not all chapters have associated code sample files.

Where to Go from Here

If you're starting from scratch with Hadoop, we recommend you start at the beginning and truck your way on through the whole book. But Hadoop does a lot of different things, so if you come to a chapter or section that covers an area you won't be using, feel free to skip it. Or if you're not a total newbie, you can bypass the parts you're familiar with. We wrote this book so that you can dive in anywhere.

If you're a selective reader and you just want to try out the examples in the book, we strongly recommend looking at Chapter 3. It's here that we describe how to set up your own Hadoop environment in a Virtual Machine (VM) that you can run on your own computer. All the examples and code samples were tested using this environment, and we've laid out all the steps you need to download, install, and configure Hadoop.

Part I

Getting Started with Hadoop

Visit `www.dummies.com` for great Dummies content online.

In this part . . .

- ✓ See what makes Hadoop-sense — and what doesn't.
- ✓ Look at what Hadoop is doing to raise productivity in the real world.
- ✓ See what's involved in setting up a Hadoop environment
- ✓ Visit `www.dummies.com` for great Dummies content online.

Chapter 1

Introducing Hadoop and Seeing What It's Good For

In This Chapter

- Seeing how Hadoop fills a need
- Digging (a bit) into Hadoop's history
- Getting Hadoop for yourself
- Looking at Hadoop application offerings

Organizations are flooded with data. Not only that, but in an era of incredibly cheap storage where everyone and everything are interconnected, the nature of the data we're collecting is also changing. For many businesses, their critical data used to be limited to their transactional databases and data warehouses. In these kinds of systems, data was organized into orderly rows and columns, where every byte of information was well understood in terms of its nature and its business value. These databases and warehouses are still extremely important, but businesses are now differentiating themselves by how they're finding value in the large volumes of data that are *not* stored in a tidy database.

The variety of data that's available now to organizations is incredible: Internally, you have website clickstream data, typed notes from call center operators, e-mail and instant messaging repositories; externally, open data initiatives from public and private entities have made massive troves of raw data available for analysis. The challenge here is that traditional tools are poorly equipped to deal with the scale and complexity of much of this data. That's where Hadoop comes in. It's tailor-made to deal with all sorts of messiness. CIOs everywhere have taken notice, and Hadoop is rapidly becoming an established platform in any serious IT department.

This chapter is a newcomer's welcome to the wonderful world of Hadoop — its design, capabilities, and uses. If you're new to big data, you'll also find important background information that applies to Hadoop and other solutions.

Big Data and the Need for Hadoop

Like many buzzwords, what people mean when they say "big data" is not always clear. This lack of clarity is made worse by IT people trying to attract attention to their own projects by labeling them as "big data," even though there's nothing big about them.

At its core, big data is simply a way of describing data problems that are unsolvable using traditional tools. To help understand the nature of big data problems, we like the "the three Vs of big data," which are a widely accepted characterization for the factors behind what makes a data challenge "big":

- **Volume:** High volumes of data ranging from dozens of terabytes, and even petabytes.
- **Variety:** Data that's organized in multiple structures, ranging from raw text (which, from a computer's perspective, has little or no discernible structure — many people call this *unstructured data*) to log files (commonly referred to as being *semistructured*) to data ordered in strongly typed rows and columns (*structured* data). To make things even more confusing, some data sets even include portions of all three kinds of data. (This is known as *multistructured* data.)
- **Velocity:** Data that enters your organization and has some kind of value for a limited window of time — a window that usually shuts well before the data has been transformed and loaded into a data warehouse for deeper analysis (for example, financial securities ticker data, which may reveal a buying opportunity, but only for a short while). The higher the volumes of data entering your organization per second, the bigger your velocity challenge.

Each of these criteria clearly poses its own, distinct challenge to someone wanting to analyze the information. As such, these three criteria are an easy way to assess big data problems and provide clarity to what has become a vague buzzword. The commonly held rule of thumb is that if your data storage and analysis work exhibits any of these three characteristics, chances are that you've got yourself a big data challenge.

Failed attempts at coolness: Naming technologies

The co-opting of the big data label reminds us when Java was first becoming popular in the early 1990s and every IT project had to have Java support or something to do with Java. At the same time, web site application development was becoming popular and Netscape named their scripting language "JavaScript," even though it had nothing to do with Java. To this day, people are confused by this shallow naming choice.

Origin of the "3 Vs"

In 2001, years before marketing people got ahold of the term "big data," the analyst firm META Group published a report titled *3-D Data Management: Controlling Data Volume, Velocity and Variety*. This paper was all about data warehousing challenges, and ways to use relational technologies to overcome them. So while the definitions of the 3Vs in this paper are quite different from the big data 3Vs, this paper does deserve a footnote in the history of big data, since it originated a catchy way to describe a problem.

As you'll see in this book, Hadoop is anything but a traditional information technology tool, and it is well suited to meet many big data challenges, especially (as you'll soon see) with high volumes of data and data with a variety of structures. But there are also big data challenges where Hadoop isn't well suited — in particular, analyzing high-velocity data the instant it enters an organization. Data velocity challenges involve the analysis of data while it's in motion, whereas Hadoop is tailored to analyze data when it's at rest. The lesson to draw from this is that although Hadoop is an important tool for big data analysis, it will by no means solve all your big data problems. Unlike some of the buzz and hype, the entire big data domain isn't synonymous with Hadoop.

Exploding data volumes

It is by now obvious that we live in an advanced state of the information age. Data is being generated and captured electronically by networked sensors at tremendous volumes, in ever-increasing velocities and in mind-boggling varieties. Devices such as mobile telephones, cameras, automobiles, televisions, and machines in industry and health care all contribute to the exploding data volumes that we see today. This data can be browsed, stored, and shared, but its greatest value remains largely untapped. That value lies in its potential to provide insight that can solve vexing business problems, open new markets, reduce costs, and improve the overall health of our societies.

In the early 2000s (we like to say "the oughties"), companies such as Yahoo! and Google were looking for a new approach to analyzing the huge amounts of data that their search engines were collecting. Hadoop is the result of that effort, representing an efficient and cost-effective way of reducing huge analytical challenges to small, manageable tasks.

Varying data structures

Structured data is characterized by a high degree of organization and is typically the kind of data you see in relational databases or spreadsheets. Because of its defined structure, it maps easily to one of the standard data types (or user-defined types that are based on those standard types). It can be searched using standard search algorithms and manipulated in well-defined ways.

Semistructured data (such as what you might see in log files) is a bit more difficult to understand than structured data. Normally, this kind of data is stored in the form of text files, where there is some degree of order — for example, tab-delimited files, where columns are separated by a tab character. So instead of being able to issue a database query for a certain column and knowing exactly what you're getting back, users typically need to explicitly assign data types to any data elements extracted from semistructured data sets.

Unstructured data has none of the advantages of having structure coded into a data set. (To be fair, the unstructured label is a bit strong — all data stored in a computer has some degree of structure. When it comes to so-called unstructured data, there's simply too little structure in order to make much sense of it.) Its analysis by way of more traditional approaches is difficult and costly at best, and logistically impossible at worst. Just imagine having many years' worth of notes typed by call center operators that describe customer observations. Without a robust set of text analytics tools, it would be extremely tedious to determine any interesting behavior patterns. Moreover, the sheer volume of data in many cases poses virtually insurmountable challenges to traditional data mining techniques, which, even when conditions are good, can handle only a fraction of the valuable data that's available.

A playground for data scientists

A *data scientist* is a computer scientist who loves data (lots of data) and the sublime challenge of figuring out ways to squeeze every drop of value out of that abundant data. A *data playground* is an enterprise store of many terabytes (or even petabytes) of data that data scientists can use to develop, test, and enhance their analytical "toys."

Now that you know what big data is all about, what it is, and why it's important, it's time to introduce Hadoop, the granddaddy of these nontraditional analytical toys. Understanding how this amazing platform for the analysis of big data came to be, and acquiring some basic principles about how it works, will help you to master the details we provide in the remainder of this book.

The Origin and Design of Hadoop

So what exactly is this thing with the funny name — Hadoop? At its core, Hadoop is a framework for storing data on large clusters of *commodity* hardware — everyday computer hardware that is affordable and easily available — and running applications against that data. A *cluster* is a group of interconnected computers (known as *nodes*) that can work together on the same problem. Using networks of affordable compute resources to acquire business insight is the key value proposition of Hadoop.

As for that name, Hadoop, don't look for any major significance there; it's simply the name that Doug Cutting's son gave to his stuffed elephant. (Doug Cutting is, of course, the co-creator of Hadoop.) The name is unique and easy to remember — characteristics that made it a great choice.

Hadoop consists of two main components: a distributed processing framework named MapReduce (which is now supported by a component called YARN, which we describe a little later) and a distributed file system known as the Hadoop distributed file system, or HDFS.

An application that is running on Hadoop gets its work divided among the nodes (machines) in the cluster, and HDFS stores the data that will be processed. A Hadoop cluster can span thousands of machines, where HDFS stores data, and MapReduce jobs do their processing near the data, which keeps I/O costs low. MapReduce is extremely flexible, and enables the development of a wide variety of applications.

TECHNICAL STUFF

As you might have surmised, a Hadoop cluster is a form of *compute cluster,* a type of cluster that's used mainly for computational purposes. In a compute cluster, many computers (*compute nodes*) can share computational workloads and take advantage of a very large aggregate bandwidth across the cluster. Hadoop clusters typically consist of a few *master nodes,* which control the storage and processing systems in Hadoop, and many *slave nodes,* which store all the cluster's data and is also where the data gets processed.

Distributed processing with MapReduce

MapReduce involves the processing of a sequence of operations on distributed data sets. The data consists of key-value pairs, and the computations have only two phases: a map phase and a reduce phase. User-defined MapReduce jobs run on the compute nodes in the cluster.

A look at the history books

Hadoop was originally intended to serve as the infrastructure for the Apache Nutch project, which started in 2002. Nutch, an open source web search engine, is a part of the Lucene project. What are these projects? Apache projects are created to develop open source software and are supported by the Apache Software Foundation (ASF), a nonprofit corporation made up of a decentralized community of developers. *Open source software,* which is usually developed in a public and collaborative way, is software whose source code is freely available to anyone for study, modification, and distribution.

Nutch needed an architecture that could scale to billions of web pages, and the needed architecture was inspired by the Google file system (GFS), and would ultimately become HDFS. In 2004, Google published a paper that introduced MapReduce, and by the middle of 2005 Nutch was using both MapReduce and HDFS.

In early 2006, MapReduce and HDFS became part of the Lucene subproject named Hadoop, and by February 2008, the Yahoo! search index was being generated by a Hadoop cluster. By the beginning of 2008, Hadoop was a top-level project at Apache and was being used by many companies. In April 2008, Hadoop broke a world record by sorting a terabyte of data in 209 seconds, running on a 910-node cluster. By May 2009, Yahoo! was able to use Hadoop to sort 1 terabyte in 62 seconds!

Generally speaking, a MapReduce job runs as follows:

1. During the Map phase, input data is split into a large number of fragments, each of which is assigned to a map task.
2. These map tasks are distributed across the cluster.
3. Each map task processes the key-value pairs from its assigned fragment and produces a set of intermediate key-value pairs.
4. The intermediate data set is sorted by key, and the sorted data is partitioned into a number of fragments that matches the number of reduce tasks.
5. During the Reduce phase, each reduce task processes the data fragment that was assigned to it and produces an output key-value pair.
6. These reduce tasks are also distributed across the cluster and write their output to HDFS when finished.

The Hadoop MapReduce framework in earlier (pre-version 2) Hadoop releases has a single master service called a JobTracker and several slave services called TaskTrackers, one per node in the cluster. When you submit a MapReduce job to the JobTracker, the job is placed into a queue and then runs according to the scheduling rules defined by an administrator. As you might expect, the JobTracker manages the assignment of map-and-reduce tasks to the TaskTrackers.

With Hadoop 2, a new resource management system is in place called YARN (short for *Y*et *A*nother *R*esource *M*anager). YARN provides generic scheduling and resource management services so that you can run more than just Map Reduce applications on your Hadoop cluster. The JobTracker/TaskTracker architecture could only run MapReduce.

We describe YARN and the JobTracker/TaskTracker architectures in Chapter 7.

HDFS also has a master/slave architecture:

- **Master service:** Called a *NameNode,* it controls access to data files.
- **Slave services:** Called *DataNodes,* they're distributed one per node in the cluster. DataNodes manage the storage that's associated with the nodes on which they run, serving client read and write requests, among other tasks.

For more information on HDFS, see Chapter 4.

Apache Hadoop ecosystem

This section introduces other open source components that are typically seen in a Hadoop deployment. Hadoop is more than MapReduce and HDFS: It's also a family of related projects (an ecosystem, really) for distributed computing and large-scale data processing. Most (but not all) of these projects are hosted by the Apache Software Foundation. Table 1-1 lists some of these projects.

Table 1-1 Related Hadoop Projects

Project Name	*Description*
Ambari	An integrated set of Hadoop administration tools for installing, monitoring, and maintaining a Hadoop cluster. Also included are tools to add or remove slave nodes.
Avro	A framework for the efficient *serialization* (a kind of transformation) of data into a compact binary format
Flume	A data flow service for the movement of large volumes of log data into Hadoop
HBase	A distributed columnar database that uses HDFS for its underlying storage. With HBase, you can store data in extremely large tables with variable column structures
HCatalog	A service for providing a relational view of data stored in Hadoop, including a standard approach for tabular data

(continued)

Table 1-1 *(continued)*

Project Name	***Description***
Hive	A distributed data warehouse for data that is stored in HDFS; also provides a query language that's based on SQL (HiveQL)
Hue	A Hadoop administration interface with handy GUI tools for browsing files, issuing Hive and Pig queries, and developing Oozie workflows
Mahout	A library of machine learning statistical algorithms that were implemented in MapReduce and can run natively on Hadoop
Oozie	A workflow management tool that can handle the scheduling and chaining together of Hadoop applications
Pig	A platform for the analysis of very large data sets that runs on HDFS and with an infrastructure layer consisting of a compiler that produces sequences of MapReduce programs and a language layer consisting of the query language named Pig Latin
Sqoop	A tool for efficiently moving large amounts of data between relational databases and HDFS
ZooKeeper	A simple interface to the centralized coordination of services (such as naming, configuration, and synchronization) used by distributed applications

The Hadoop ecosystem and its commercial distributions (see the "Comparing distributions" section, later in this chapter) continue to evolve, with new or improved technologies and tools emerging all the time.

Figure 1-1 shows the various Hadoop ecosystem projects and how they relate to one-another:

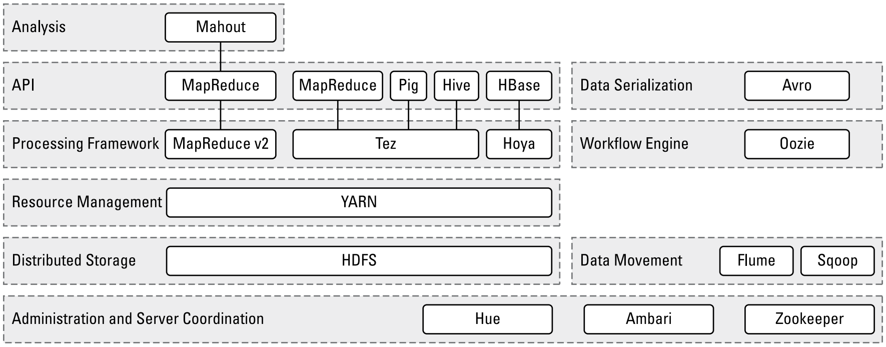


Figure 1-1: Hadoop ecosystem components.