Community Experience Distilled

Hadoop Essentials

Delve into the key concepts of Hadoop and get a thorough understanding of the Hadoop ecosystem

Shiva Achari

[PACKT] PUBLISHING

Hadoop Essentials

Delve into the key concepts of Hadoop and get a thorough understanding of the Hadoop ecosystem

Shiva Achari

[PACKT]
PUBLISHING

BIRMINGHAM - MUMBAI

Hadoop Essentials

First published: April 2015

Production reference: 1240415

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78439-668-8

www.packtpub.com

Credits

Author
Shiva Achari

Reviewers
Anindita Basak
Ralf Becher
Marius Danciu
Dmitry Spikhalskiy

Commissioning Editor
Sarah Crofton

Acquisition Editor
Subho Gupta

Content Development Editor
Rahul Nair

Technical Editor
Bharat Patil

Copy Editors
Hiral Bhat
Charlotte Carneiro
Puja Lalwani
Sonia Mathur
Kriti Sharma
Sameen Siddiqui

Project Coordinator
Leena Purkait

Proofreaders
Simran Bhogal
Safis Editing
Linda Morris

Indexer
Priya Sane

Graphics
Sheetal Aute
Jason Monteiro

Production Coordinator
Komal Ramchandani

Cover Work
Komal Ramchandani

About the Author

Shiva Achari has over 8 years of extensive industry experience and is currently working as a Big Data Architect consultant with companies such as Oracle and Teradata. Over the years, he has architected, designed, and developed multiple innovative and high-performance large-scale solutions, such as distributed systems, data centers, big data management tools, SaaS cloud applications, Internet applications, and Data Analytics solutions.

He is also experienced in designing big data and analytics applications, such as ingestion, cleansing, transformation, correlation of different sources, data mining, and user experience in Hadoop, Cassandra, Solr, Storm, R, and Tableau.

He specializes in developing solutions for the big data domain and possesses sound hands-on experience on projects migrating to the Hadoop world, new developments, product consulting, and POC. He also has hands-on expertise in technologies such as Hadoop, Yarn, Sqoop, Hive, Pig, Flume, Solr, Lucene, Elasticsearch, Zookeeper, Storm, Redis, Cassandra, HBase, MongoDB, Talend, R, Mahout, Tableau, Java, and J2EE.

He has been involved in reviewing *Mastering Hadoop*, *Packt Publishing*.

Shiva has expertise in requirement analysis, estimations, technology evaluation, and system architecture along with domain experience in telecoms, Internet applications, document management, healthcare, and media.

Currently, he is supporting presales activities such as writing technical proposals (RFP), providing technical consultation to customers, and managing deliveries of big data practice groups in Teradata.

He is active on his LinkedIn page at `http://in.linkedin.com/in/shivaachari/`.

Acknowledgments

I would like to dedicate this book to my family, especially my father, mother, and wife. My father is my role model and I cannot find words to thank him enough, and I'm missing him as he passed away last year. My wife and mother have supported me throughout my life. I'd also like to dedicate this book to a special one whom we are expecting this July. Packt Publishing has been very kind and supportive, and I would like to thank all the individuals who were involved in editing, reviewing, and publishing this book. Some of the content was taken from my experiences, research, studies, and from the audiences of some of my trainings. I would like to thank my audience who found the book worth reading and hope that you gain the knowledge and help and implement them in your projects.

About the Reviewers

Anindita Basak is working as a big data cloud consultant and trainer and is highly enthusiastic about core Apache Hadoop, vendor-specific Hadoop distributions, and the Hadoop open source ecosystem. She works as a specialist in a big data start-up in the Bay area and with fortune brand clients across the U.S. She has been playing with Hadoop on Azure from the days of its incubation (that is, `www.hadooponazure.com`). Previously in her role, she has worked as a module lead for Alten Group Company and in the Azure Pro Direct Delivery group for Microsoft. She has also worked as a senior software engineer on the implementation and migration of various enterprise applications on Azure Cloud in the healthcare, retail, and financial domain. She started her journey with Microsoft Azure in the Microsoft Cloud Integration Engineering (CIE) team and worked as a support engineer for Microsoft India (R&D) Pvt. Ltd.

With more than 7 years of experience with the Microsoft .NET, Java, and the Hadoop technology stack, she is solely focused on the big data cloud and data science. She is a technical speaker, active blogger, and conducts various training programs on the Hortonworks and Cloudera developer/administrative certification programs. As an MVB, she loves to share her technical experience and expertise through her blog at `http://anindita9.wordpress.com` and `http://anindita9.azurewebsites.net`. You can get a deeper insight into her professional life on her LinkedIn page, and you can follow her on Twitter. Her Twitter handle is `@imcuteani`.

She recently worked as a technical reviewer for *HDInsight Essentials (volume I and II)* and *Microsoft Tabular Modeling Cookbook*, both by Packt Publishing.

Ralf Becher has worked as an IT system architect and data management consultant for more than 15 years in the areas of banking, insurance, logistics, automotive, and retail.

He is specialized in modern, quality-assured data management. He has been helping customers process, evaluate, and maintain the quality of the company data by helping them introduce, implement, and improve complex solutions in the fields of data architecture, data integration, data migration, master data management, metadata management, data warehousing, and business intelligence.

He started working with big data on Hadoop in 2012. He runs his BI and data integration blog at `http://irregular-bi.tumblr.com/`.

Marius Danciu has over 15 years of experience in developing and architecting Java platform server-side applications in the data synchronization and big data analytics fields. He's very fond of the Scala programming language and functional programming concepts and finding its applicability in everyday work. He is the coauthor of *The Definitive Guide to Lift*, *Apress*.

Dmitry Spikhalskiy is currently holding the position of a software engineer at the Russian social network, Odnoklassniki, and working on a search engine, video recommendation system, and movie content analysis.

Previously, he took part in developing the Mind Labs' platform and its infrastructure, and benchmarks for high load video conference and streaming services, which got "The biggest online-training in the world" Guinness World Record. More than 12,000 people participated in this competition. He also a mobile social banking start-up called Instabank as its technical lead and architect. He has also reviewed *Learning Google Guice, PostgreSQL 9 Admin Cookbook*, and *Hadoop MapReduce v2 Cookbook*, all by Packt Publishing.

He graduated from Moscow State University with an MSc degree in computer science, where he first got interested in parallel data processing, high load systems, and databases.

www.PacktPub.com

Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit `www.PacktPub.com`.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at `www.PacktPub.com` and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `service@packtpub.com` for more details.

At `www.PacktPub.com`, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

`https://www2.packtpub.com/books/subscription/packtlib`

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

Free access for Packt account holders

If you have an account with Packt at `www.PacktPub.com`, you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface

Hadoop is quite a fascinating and interesting project that has seen quite a lot of interest and contributions from the various organizations and institutions. Hadoop has come a long way, from being a batch processing system to a data lake and high-volume streaming analysis in low latency with the help of various Hadoop ecosystem components, specifically YARN. This progress has been substantial and has made Hadoop a powerful system, which can be designed as a storage, transformation, batch processing, analytics, or streaming and real-time processing system.

Hadoop project as a data lake can be divided in multiple phases such as data ingestion, data storage, data access, data processing, and data management. For each phase, we have different sub-projects that are tools, utilities, or frameworks to help and accelerate the process. The Hadoop ecosystem components are tested, configurable and proven and to build similar utility on our own it would take a huge amount of time and effort to achieve. The core of the Hadoop framework is complex for development and optimization. The smart way to speed up and ease the process is to utilize different Hadoop ecosystem components that are very useful, so that we can concentrate more on the application flow design and integration with other systems.

With the emergence of many useful sub-projects in Hadoop and other tools within the Hadoop ecosystem, the question that arises is which tool to use when and how effectively. This book is intended to complete the jigsaw puzzle of when and how to use the various ecosystem components, and to make you well aware of the Hadoop ecosystem utilities and the cases and scenarios where they should be used.

What this book covers

Chapter 1, *Introduction to Big Data and Hadoop*, covers an overview of big data and Hadoop, plus different use case patterns with advantages and features of Hadoop.

Chapter 2, *Hadoop Ecosystem*, explores the different phases or layers of Hadoop project development and some components that can be used in each layer.

Chapter 3, *Pillars of Hadoop – HDFS, MapReduce, and YARN*, is about the three key basic components of Hadoop, which are HDFS, MapReduce, and YARN.

Chapter 4, *Data Access Components – Hive and Pig*, covers the data access components Hive and Pig, which are abstract layers of the SQL-like and Pig Latin procedural languages, respectively, on top of the MapReduce framework.

Chapter 5, *Storage Components – HBase*, is about the NoSQL component database HBase in detail.

Chapter 6, *Data Ingestion in Hadoop – Sqoop and Flume*, covers the data ingestion library tools Sqoop and Flume.

Chapter 7, *Streaming and Real-time Analysis – Storm and Spark*, is about the streaming and real-time frameworks Storm and Spark built on top of YARN.

What you need for this book

A prerequisite for this book is good understanding of Java programming and basics of distributed computing will be very helpful and an interest to understand about Hadoop and its ecosystem components.

The code and syntax have been tested in Hadoop 2.4.1 and other compatible ecosystem component versions, but may vary in the newer version.

Who this book is for

If you are a system or application developer interested in learning how to solve practical problems using the Hadoop framework, then this book is ideal for you. This book is also meant for Hadoop professionals who want to find solutions to the different challenges they come across in their Hadoop projects. It assumes a familiarity with distributed storage and distributed applications.

Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "We can include other contexts through the use of the `include` directive."

A block of code is set as follows:

```
public static class MyPartitioner extends   org.apache.hadoop.
mapreduce.Partitioner<Text,Text>

{
  @Override
  public int getPartition(Text key, Text value, int numPartitions)
  {
   int count =Integer.parseInt(line[1]);
   if(count<=3)
    return 0;
   else
    return 1;
  }
}

And in Driver class
job.setPartitionerClass(MyPartitioner.class);
```

Any command-line input or output is written as follows:

```
hadoop fs -put /home/shiva/Samplefile.txt  /user/shiva/dir3/
```

Warnings or important notes appear in a box like this.

Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail `feedback@packtpub.com`, and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at `www.packtpub.com/authors`.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files from your account at `http://www.packtpub.com` for all the Packt Publishing books you have purchased. If you purchased this book elsewhere, you can visit `http://www.packtpub.com/support` and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/submit-errata`, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to `https://www.packtpub.com/books/content/support` and enter the name of the book in the search field. The required information will appear under the **Errata** section.

Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

Questions

If you have a problem with any aspect of this book, you can contact us at `questions@packtpub.com`, and we will do our best to address the problem.

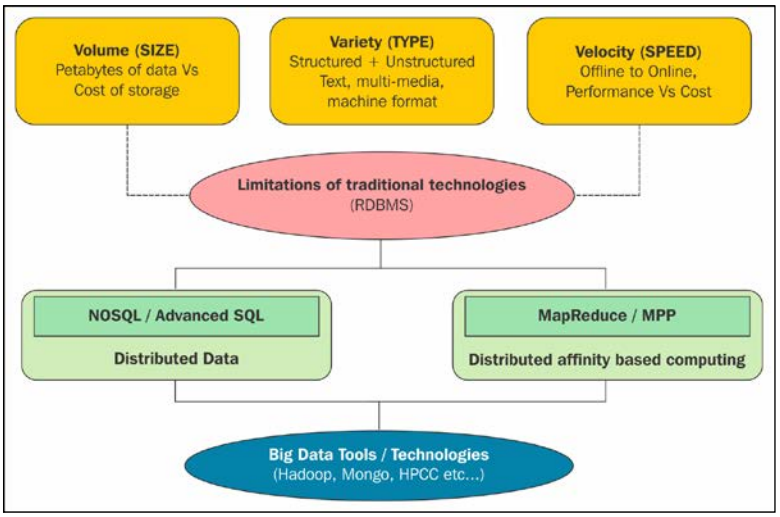
1
Introduction to Big Data and Hadoop

Hello big data enthusiast! By this time, I am sure you must have heard a lot about big data, as big data is the hot IT buzzword and there is a lot of excitement about big data. Let us try to understand the necessities of big data. There are humungous amount of data, available on the Internet, at institutions, and with some organizations, which have a lot of meaningful insights, which can be analyzed using data science techniques and involves complex algorithms. Data science techniques require a lot of processing time, intermediate data(s), and CPU power, that may take roughly tens of hours on gigabytes of data and data science works on a trial and error basis, to check if an algorithm can process the data better or not to get such insights. Big data systems can process data analytics not only faster but also efficiently for a large data and can enhance the scope of R&D analysis and can yield more meaningful insights and faster than any other analytic or BI system.

Big data systems have emerged due to some issues and limitations in traditional systems. The traditional systems are good for **Online Transaction Processing** (**OLTP**) and **Business Intelligence** (**BI**), but are not easily scalable considering cost, effort, and manageability aspect. Processing heavy computations are difficult and prone to memory issues, or will be very slow, which hinders data analysis to a greater extent. Traditional systems lack extensively in data science analysis and make big data systems powerful and interesting. Some examples of big data use cases are predictive analytics, fraud analytics, machine learning, identifying patterns, data analytics, semi-structured, and unstructured data processing and analysis.

V's of big data

Typically, the problem that comes in the bracket of big data is defined by terms that are often called as V's of big data. There are typically three V's, which are Volume, Velocity, and variety, as shown in the following image:



Volume

According to the fifth annual survey by **International Data Corporation** (**IDC**), 1.8 zettabytes (1.8 trillion gigabytes) of information were created and replicated in 2011 alone, which is up from 800 GB in 2009, and the number is expected to more than double every two years surpassing 35 zettabytes by 2020. Big data systems are designed to store these amounts of data and even beyond that too with a fault tolerant architecture, and as it is distributed and replicated across multiple nodes, the underlying nodes can be average computing systems, which too need not be high performing systems, which reduces the cost drastically.

The cost per terabyte storage in big data is very less than in other systems, and this has made organizations interested to a greater extent, and even if the data grows multiple times, it is easily scalable, and nodes can be added without much maintenance effort.

Velocity

Processing and analyzing the amount of data that we discussed earlier is one of the key interest areas where big data is gaining popularity and has grown enormously. Not all data to be processed has to be larger in volume initially, but as we process and execute some complex algorithms, the data can grow massively. For processing most of the algorithms, we would require intermediate or temporary data, which can be in GB or TB for big data, so while processing, we would require some significant amount of data, and processing also has to be faster. Big data systems can process huge complex algorithms on huge data much quickly, as it leverages parallel processing across distributed environment, which executes multiple processes in parallel at the same time, and the job can be completed much faster.

For example, Yahoo created a world record in 2009 using Apache Hadoop for sorting a petabyte in 16.25 hours and a terabyte in 62 seconds. MapR have achieved terabyte data sorting in 55 seconds, which speaks volume for the processing power, especially in analytics where we need to use a lot of intermediate data to perform heavy time and memory intensive algorithms much faster.

Variety

Another big challenge for the traditional systems is to handle different variety of semi-structured data or unstructured data such as e-mails, audio and video analysis, image analysis, social media, gene, geospatial, 3D data, and so on. Big data can not only help store, but also utilize and process such data using algorithms much more quickly and also efficiently. Semi-structured and unstructured data processing is complex, and big data can use the data with minimal or no preprocessing like other systems and can save a lot of effort and help minimize loss of data.

Understanding big data

Actually, big data is a terminology which refers to challenges that we are facing due to exponential growth of data in terms of V problems. The challenges can be subdivided into the following phases:

- Capture
- Storage
- Search
- Sharing
- Analytics
- Visualization