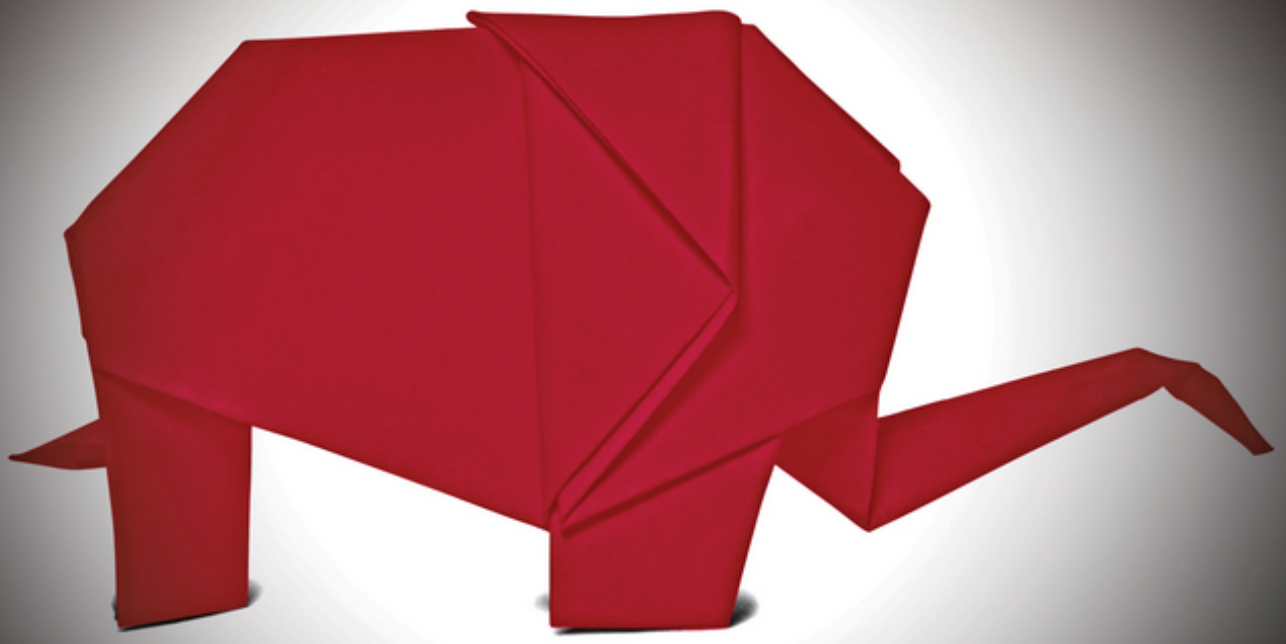

Professional Hadoop® Solutions

Boris Lublinsky, Kevin T. Smith, Alexey Yakubovich

1

Big Data and the Hadoop Ecosystem

WHAT'S IN THIS CHAPTER?

- Understanding the challenges of Big Data
- Getting to know the Hadoop ecosystem
- Getting familiar with Hadoop distributions
- Using Hadoop-based enterprise applications

Everyone says it — we are living in the era of "Big Data." Chances are that you have heard this phrase. In today's technology-fueled world where computing power has significantly increased, electronic devices are more commonplace, accessibility to the Internet has improved, and users have been able to transmit and collect more data than ever before.

Organizations are producing data at an astounding rate. It is reported that Facebook alone collects 250 terabytes a day. According to Thompson Reuters News Analytics, digital data production has more than doubled from almost 1 million petabytes (equal to about 1 billion terabytes) in 2009 to a projected 7.9 zettabytes (a zettabyte is equal to 1 million petabytes) in 2015, and an estimated 35 zettabytes in 2020. Other research organizations offer even higher estimates!

As organizations have begun to collect and produce massive amounts of data, they have recognized the advantages of data analysis. But they have also struggled to manage the massive amounts of information that they have. This has led to new challenges. How can you effectively store such a massive quantity of data? How can you effectively process it? How can you analyze your data in an efficient manner? Knowing that data will only increase, how can you build a solution that will scale?

These challenges that come with Big Data are not just for academic researchers and data scientists. In a Google+ conversation a few years ago, noted computer book publisher Tim O'Reilly made a point of quoting Alistair Croll, who said that "companies that have massive amounts of data without massive amounts of clue are going to be displaced by startups that have less data but more clue …" In short, what Croll was saying was that unless your business *understands* the data it has, it will not be able to compete with businesses that do.

Businesses realize that tremendous benefits can be gained in analyzing Big Data related to business competition, situational awareness, productivity, science, and innovation. Because competition is driving the analysis of Big Data, most organizations agree with O'Reilly and Croll. These organizations believe that the survival of today's companies will depend on their capability to store, process, and analyze massive amounts of information, and to master the Big Data challenges.

If you are reading this book, you are most likely familiar with these challenges, you have some familiarity with Apache Hadoop, and you know that Hadoop can be used to solve these problems. This chapter explains the promises and the challenges of Big Data. It also provides a high-level overview of Hadoop and its ecosystem of software components that can be used together to build scalable, distributed data analytics solutions.

BIG DATA MEETS HADOOP

Citing "human capital" as an intangible but crucial element of their success, most organizations will suggest that their employees are their most valuable asset. Another critical asset that is typically not listed on a corporate balance sheet is the *information* that a company has. The power of an organization's information can be enhanced by its trustworthiness, its volume, its accessibility, and the capability of an organization to be able to make sense of it all in a reasonable amount of time in order to empower intelligent decision making.

It is very difficult to comprehend the sheer amount of digital information that organizations produce. IBM states that 90 percent of the digital data in the world was created in the past two years alone. Organizations are collecting, producing, and storing this data, which can be a strategic resource. A book written more than a decade ago, *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management* by Michael Daconta, Leo Obrst, and Kevin T. Smith (Indianapolis: Wiley, 2004) included a maxim that said, "The organization that has the best information, knows how to find it, and can utilize it the quickest wins."

Knowledge is power. The problem is that with the vast amount of digital information being collected, traditional database tools haven't been able to manage or process this information quickly enough. As a result, organizations have been drowning in data. Organizations haven't been able to use the data well, and haven't been able to "connect the dots" in the data quickly enough to understand the power in the information that the data presents.

The term "Big Data" has been used to describe data sets that are so large that typical and traditional means of data storage, management, search, analytics, and other processing has become a challenge. Big Data is characterized by the magnitude of digital information that can come from many sources and data formats (structured and unstructured), and data that can be processed and analyzed to find insights and patterns used to make informed decisions.

What are the challenges with Big Data? How can you store, process, and analyze such a large amount of data to identify patterns and knowledge from a massive sea of information?

Analyzing Big Data requires lots of storage and large computations that demand a great deal of processing power. As digital information began to increase over the past decade, organizations tried different approaches to solving these problems. At first, focus was placed on giving individual machines more storage, processing power, and memory — only to quickly find that analytical techniques on single machines failed to scale. Over time, many realized the promise of distributed systems (distributing tasks over multiple machines), but data analytic solutions were often complicated, error-prone, or simply not fast enough.

In 2002, while developing a project called Nutch (a search engine project focused on crawling, indexing, and searching Internet web pages), Doug Cutting and Mike Cafarella were struggling with a solution for processing a vast amount of information. Realizing the storage and processing demands for Nutch, they knew that they would need a reliable, distributed computing approach that would scale to the demand of the vast amount of website data that the tool would be collecting.

A year later, Google published papers on the Google File System (GFS) and MapReduce, an algorithm and distributed programming platform for processing large data sets. Recognizing the promise of these approaches used by Google for distributed processing and storage over a cluster of machines, Cutting and Cafarella used this work as the basis of building the distributed platform for Nutch, resulting in what we now know as the Hadoop Distributed File System (HDFS) and Hadoop's implementation of MapReduce.

In 2006, after struggling with the same "Big Data" challenges related to indexing massive amounts of information for its search engine, and after watching the progress of the Nutch project, Yahoo! hired Doug Cutting, and quickly decided to adopt Hadoop as its distributed framework for solving its search engine challenges. Yahoo! spun out the storage and processing parts of Nutch to form Hadoop as an open source Apache project, and the Nutch web crawler remained its own separate project. Shortly thereafter, Yahoo! began rolling out Hadoop as a means to power analytics for various production applications. The platform was so effective that Yahoo! merged its search and advertising into one unit to better leverage Hadoop technology.

In the past 10 years, Hadoop has evolved from its search engine–related origins to one of the most popular general-purpose computing platforms for solving Big Data challenges. It is quickly becoming the foundation for the next generation of data-based applications. The market research firm IDC predicts that Hadoop will be driving a Big Data market that should hit more than $23 billion by 2016. Since the launch of the first Hadoop-centered company, Cloudera, in 2008, dozens of Hadoop-based startups have attracted hundreds of millions of dollars in venture capital investment. Simply put, organizations have found that Hadoop offers a proven approach to Big Data analytics.

Hadoop: Meeting the Big Data Challenge

Apache Hadoop meets the challenges of Big Data by simplifying the implementation of data-intensive, highly parallel distributed applications. Used throughout the world by businesses, universities, and other organizations, it allows analytical tasks to be divided into fragments of work and distributed over thousands of computers, providing fast analytics time and distributed storage

of massive amounts of data. Hadoop provides a cost-effective way for storing huge quantities of data. It provides a scalable and reliable mechanism for processing large amounts of data over a cluster of commodity hardware. And it provides new and improved analysis techniques that enable sophisticated analytical processing of multi-structured data.

Hadoop is different from previous distributed approaches in the following ways:

- Data is distributed in advance.
- Data is replicated throughout a cluster of computers for reliability and availability.
- Data processing tries to occur where the data is stored, thus eliminating bandwidth bottlenecks.

In addition, Hadoop provides a simple programming approach that abstracts the complexity evident in previous distributed implementations. As a result, Hadoop provides a powerful mechanism for data analytics, which consists of the following:

- **Vast amount of storage** — Hadoop enables applications to work with thousands of computers and petabytes of data. Over the past decade, computer professionals have realized that low-cost "commodity" systems can be used together for high-performance computing applications that once could be handled only by supercomputers. Hundreds of "small" computers may be configured in a cluster to obtain aggregate computing power that can exceed by far that of single supercomputer at a cheaper price. Hadoop can leverage clusters in excess of thousands of machines, providing huge storage and processing power at a price that an enterprise can afford.
- **Distributed processing with fast data access** — Hadoop clusters provide the capability to efficiently store vast amounts of data while providing fast data access. Prior to Hadoop, parallel computation applications experienced difficulty distributing execution between machines that were available on the cluster. This was because the cluster execution model creates demand for shared data storage with very high I/O performance. Hadoop moves execution toward the data. Moving the applications to the data alleviates many of the high-performance challenges. In addition, Hadoop applications are typically organized in a way that they process data sequentially. This avoids random data access (disk seek operations), further decreasing I/O load.
- **Reliability, failover, and scalability** — In the past, implementers of parallel applications struggled to deal with the issue of reliability when it came to moving to a cluster of machines. Although the reliability of an individual machine is fairly high, the probability of failure grows as the size of the cluster grows. It will not be uncommon to have daily failures in a large (thousands of machines) cluster. Because of the way that Hadoop was designed and implemented, a failure (or set of failures) will not create inconsistent results. Hadoop detects failures and retries execution (by utilizing different nodes). Moreover, the scalability support built into Hadoop's implementation allows for seamlessly bringing additional (repaired) servers into a cluster, and leveraging them for both data storage and execution.

For most Hadoop users, the most important feature of Hadoop is the clean separation between business programming and infrastructure support. For users who want to concentrate on business logic, Hadoop hides infrastructure complexity, and provides an easy-to-use platform for making complex, distributed computations for difficult problems.

Data Science in the Business World

The capability of Hadoop to store and process huge amounts of data is frequently associated with "data science." Although the term was introduced by Peter Naur in the 1960s, it did not get wide acceptance until recently. Jeffrey Stanton of Syracuse University defines it as "an emerging area of work concerned with the collection, preparation, analysis, visualization, management, and preservation of large collections of information."

Unfortunately, in business, the term is often used interchangeably with business analytics. In reality, the two disciplines are quite different.

Business analysts study patterns in existing business operations to improve them.

The goal of data science is to extract meaning from data. The work of data scientists is based on math, statistical analysis, pattern recognition, machine learning, high-performance computing, data warehousing, and much more. They analyze information to look for trends, statistics, and new business possibilities based on collected information.

Over the past few years, many business analysts more familiar with databases and programming have become data scientists, using higher-level SQL-based tools in the Hadoop ecosystem (such as Hive or real-time Hadoop queries), and running analytics to make informed business decisions.

NOT JUST "ONE BIG DATABASE"

You learn more about this a little later in this book, but before getting too far, let's dispel the notion that Hadoop is simply "one big database" meant only for data analysts. Because some of Hadoop's tools (such as Hive and real-time Hadoop queries) provide a low entry barrier to Hadoop for people more familiar with database queries, some people limit their knowledge to only a few database-centric tools in the Hadoop ecosystem.

Moreover, if the problem that you are trying to solve goes beyond data analytics and involves true "data science" problems, data mining SQL is becoming significantly less useful. Most of these problems, for example, require linear algebra, and other complex mathematical applications that are not well-translated into SQL.

This means that, although important, SQL-based tools is only one of the ways to use Hadoop. By utilizing Hadoop's MapReduce programming model, you can not only solve data science problems, but also significantly simplify enterprise application creation and deployment. You have multiple ways to do that — and you can use multiple tools, which often must be combined with other capabilities that require software-development skills. For example, by using Oozie-based application coordination (you will learn more about Oozie later in this book), you can simplify the bringing of multiple applications together, and chaining jobs from multiple tools in a very flexible way. Throughout this book, you will see practical tips for using Hadoop in your enterprise, as well as tips on when to use the right tools for the right situation.

Current Hadoop development is driven by a goal to better support data scientists. Hadoop provides a powerful computational platform, providing highly scalable, parallelizable execution that is well-suited for the creation of a new generation of powerful data science and enterprise applications. Implementers can leverage both scalable distributed storage and MapReduce processing. Businesses are using Hadoop for solving business problems, with a few notable examples:

- **Enhancing fraud detection for banks and credit card companies** — Companies are utilizing Hadoop to detect transaction fraud. By providing analytics on large clusters of commodity hardware, banks are using Hadoop, applying analytic models to a full set of transactions for their clients, and providing near-real-time fraud-in-progress detection.
- **Social media marketing analysis** — Companies are currently using Hadoop for brand management, marketing campaigns, and brand protection. By monitoring, collecting, and aggregating data from various Internet sources such as blogs, boards, news feeds, tweets, and social media, companies are using Hadoop to extract and aggregate information about their products, services, and competitors, discovering patterns and revealing upcoming trends important for understanding their business.
- **Shopping pattern analysis for retail product placement** — Businesses in the retail industry are using Hadoop to determine products most appropriate to sell in a particular store based on the store's location and the shopping patterns of the population around it.
- **Traffic pattern recognition for urban development** — Urban development often relies on traffic patterns to determine requirements for road network expansion. By monitoring traffic during different times of the day and discovering patterns, urban developers can determine traffic bottlenecks, which allow them to decide whether additional streets/street lanes are required to avoid traffic congestions during peak hours.
- **Content optimization and engagement** — Companies are focusing on optimizing content for rendering on different devices supporting different content formats. Many media companies require that a large amount of content be processed in different formats. Also, content engagement models must be mapped for feedback and enhancements.
- **Network analytics and mediation** — Real-time analytics on a large amount of data generated in the form of usage transaction data, network performance data, cell-site information, device-level data, and other forms of back office data is allowing companies to reduce operational expenses, and enhance the user experience on networks.
- **Large data transformation** — The *New York Times* needed to generate PDF files for 11 million articles (every article from 1851 to 1980) in the form of images scanned from the original paper. Using Hadoop, the newspaper was able to convert 4 TB of scanned articles to 1.5 TB of PDF documents in 24 hours.

The list of these examples could go on and on. Businesses are using Hadoop for strategic decision making, and they are starting to use their data wisely. As a result, data science has entered the business world.

BIG DATA TOOLS — NOT JUST FOR BUSINESS

Although most of the examples here focus on business, Hadoop is also widely used in the scientific community and in the public sector.

In a recent study by the Tech America Foundation, it was noted that medical researchers have demonstrated that Big Data analytics can be used to aggregate information from cancer patients to increase treatment efficacy. Police departments are using Big Data tools to develop predictive models about when and where crimes are likely to occur, decreasing crime rates. That same survey showed that energy officials are utilizing Big Data tools to analyze data related to energy consumption and potential power grid failure problems.

The bottom line is that Big Data analytics are being used to discover patterns and trends, and are used to increase efficiency and empower decision making in ways never before possible.

THE HADOOP ECOSYSTEM

When architects and developers discuss software, they typically immediately qualify a software tool for its specific usage. For example, they may say that Apache Tomcat is a web server and that MySQL is a database.

When it comes to Hadoop, however, things become a little bit more complicated. Hadoop encompasses a multiplicity of tools that are designed and implemented to work together. As a result, Hadoop can be used for many things, and, consequently, people often define it based on the way they are using it.

For some people, Hadoop is a data management system bringing together massive amounts of structured and unstructured data that touch nearly every layer of the traditional enterprise data stack, positioned to occupy a central place within a data center. For others, it is a massively parallel execution framework bringing the power of supercomputing to the masses, positioned to fuel execution of enterprise applications. Some view Hadoop as an open source community creating tools and software for solving Big Data problems. Because Hadoop provides such a wide array of capabilities that can be adapted to solve many problems, many consider it to be a basic framework.

Certainly, Hadoop provides all of these capabilities, but Hadoop should be classified as an ecosystem comprised of many components that range from data storage, to data integration, to data processing, to specialized tools for data analysts.

HADOOP CORE COMPONENTS

Although the Hadoop ecosystem is certainly growing, Figure 1-1 shows the core components.

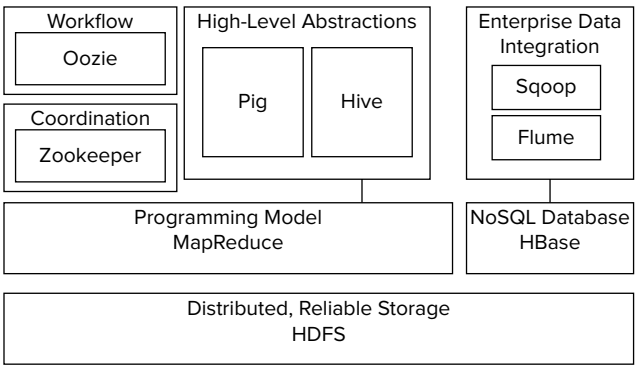

FIGURE 1-1: Core components of the Hadoop ecosystem

Starting from the bottom of the diagram in Figure 1-1, Hadoop's ecosystem consists of the following:

- **HDFS** — A foundational component of the Hadoop ecosystem is the Hadoop Distributed File System (HDFS). HDFS is the mechanism by which a large amount of data can be distributed over a cluster of computers, and data is written once, but read many times for analytics. It provides the foundation for other tools, such as HBase.
- **MapReduce** — Hadoop's main execution framework is MapReduce, a programming model for distributed, parallel data processing, breaking jobs into *mapping* phases and *reduce* phases (thus the name). Developers write *MapReduce jobs* for Hadoop, using data stored in HDFS for fast data access. Because of the nature of how MapReduce works, Hadoop brings the processing to the data in a parallel fashion, resulting in fast implementation.
- **HBase** — A column-oriented NoSQL database built on top of HDFS, HBase is used for fast read/write access to large amounts of data. HBase uses Zookeeper for its management to ensure that all of its components are up and running.
- **Zookeeper** — Zookeeper is Hadoop's distributed coordination service. Designed to run over a cluster of machines, it is a highly available service used for the management of Hadoop operations, and many components of Hadoop depend on it.
- **Oozie** — A scalable workflow system, Oozie is integrated into the Hadoop stack, and is used to coordinate execution of multiple MapReduce jobs. It is capable of managing a significant amount of complexity, basing execution on external events that include timing and presence of required data.
- **Pig** — An abstraction over the complexity of MapReduce programming, the Pig platform includes an execution environment and a scripting language (Pig Latin) used to analyze Hadoop data sets. Its compiler translates Pig Latin into sequences of MapReduce programs.
- **Hive** — An SQL-like, high-level language used to run queries on data stored in Hadoop, Hive enables developers not familiar with MapReduce to write data queries that are translated into MapReduce jobs in Hadoop. Like Pig, Hive was developed as an abstraction layer, but geared more toward database analysts more familiar with SQL than Java programming.

The Hadoop ecosystem also contains several frameworks for integration with the rest of the enterprise:

- *Sqoop* is a connectivity tool for moving data between relational databases and data warehouses and Hadoop. Sqoop leverages database to describe the schema for the imported/exported data and MapReduce for parallelization operation and fault tolerance.
- *Flume* is a distributed, reliable, and highly available service for efficiently collecting, aggregating, and moving large amounts of data from individual machines to HDFS. It is based on a simple and flexible architecture, and provides a streaming of data flows. It leverages a simple extensible data model, allowing you to move data from multiple machines within an enterprise into Hadoop.

Beyond the core components shown in Figure 1-1, Hadoop's ecosystem is growing to provide newer capabilities and components, such as the following:

- **Whirr** — This is a set of libraries that allows users to easily spin-up Hadoop clusters on top of Amazon EC2, Rackspace, or any virtual infrastructure.
- **Mahout** — This is a machine-learning and data-mining library that provides MapReduce implementations for popular algorithms used for clustering, regression testing, and statistical modeling.
- **BigTop** — This is a formal process and framework for packaging and interoperability testing of Hadoop's sub-projects and related components.
- **Ambari** — This is a project aimed at simplifying Hadoop management by providing support for provisioning, managing, and monitoring Hadoop clusters.

More members of the Hadoop family are added daily. Just during the writing of this book, three new Apache Hadoop incubator projects were added!

THE EVOLUTION OF PROJECTS INTO APACHE

If you are new to the way that the Apache Software Foundation works, and were wondering about the various projects and their relationships to each other, Apache supports the creation, maturation, and retirement of projects in an organized way. Individuals make up the membership of Apache, and together they make up the governance of the organization.

Projects start as "incubator" projects. The Apache Incubator was created to help new projects join Apache. It provides governance and reviews, and "filters" proposals to create new projects and sub-projects of existing projects. The Incubator aids in the creation of the incubated project, it evaluates the maturity of projects, and is responsible for "graduating" projects from the Incubator into Apache projects or sub-projects. The Incubator also retires projects from incubation for various reasons.

To see a full list of projects in the Incubator (current, graduated, dormant, and retired), see `http://incubator.apache.org/projects/index.html`.

The majority of Hadoop publications today either concentrate on the description of individual components of this ecosystem, or on the approach for using business analysis tools (such as Pig and Hive) in Hadoop. Although these topics are important, they typically fall short in providing an in-depth picture for helping architects build Hadoop-based enterprise applications or complex analytics applications.

HADOOP DISTRIBUTIONS

Although Hadoop is a set of open source Apache (and now GitHub) projects, a large number of companies are currently emerging with the goal of helping people actually use Hadoop. Most of these companies started with packaging Apache Hadoop distributions, ensuring that all the software worked together, and providing support. And now they are developing additional tools to simplify Hadoop usage and extend its functionality. Some of these extensions are proprietary and serve as differentiation. Some became the foundation of new projects in the Apache Hadoop family. And some are open source GitHub projects with an Apache 2 license. Although all of these companies started from the Apache Hadoop distribution, they all have a slightly different vision of what Hadoop really is, which direction it should take, and how to accomplish it.

One of the biggest differences between these companies is the use of Apache code. With the exception of the MapR, everyone considers Hadoop to be defined by the code produced by Apache projects. In contrast, MapR considers Apache code to be a reference implementation, and produces its own implementation based on the APIs provided by Apache. This approach has allowed MapR to introduce many innovations, especially around HDFS and HBase, making these two fundamental Hadoop storage mechanisms much more reliable and high-performing. Its distribution additionally introduced high-speed Network File System (NFS) access to HDFS that significantly simplifies integration of Hadoop with other enterprise applications.

Two interesting Hadoop distributions were released by Amazon and Microsoft. Both provide a prepackaged version of Hadoop running in the corresponding cloud (Amazon or Azure) as Platform as a Service (PaaS). Both provide extensions that allow developers to utilize not only Hadoop's native HDFS, but also the mapping of HDFS to their own data storage mechanisms (S3 in the case of Amazon, and Windows Azure storage in the case of Azure). Amazon also provides the capability to save and restore HBase content to and from S3.

Table 1-1 shows the main characteristics of major Hadoop distributions.

TABLE 1-1: Different Hadoop Vendors

VENDOR	HADOOP CHARACTERISTICS
Cloudera CDH, Manager, and Enterprise	Based on Hadoop 2, CDH (version 4.1.2 as of this writing) includes HDFS, YARN, HBase, MapReduce, Hive, Pig, Zookeeper, Oozie, Mahout, Hue, and other open source tools (including the real-time query engine — Impala). Cloudera Manager Free Edition includes all of CDH, plus a basic Manager supporting up to 50 cluster nodes. Cloudera Enterprise combines CDH with a more sophisticated Manager supporting an unlimited number of cluster nodes, proactive monitoring, and additional data analysis tools.

Hortonworks Data Platform	Based on Hadoop 2, this distribution (Version 2.0 Alpha as of this writing) includes HDFS, YARN, HBase, MapReduce, Hive, Pig, HCatalog, Zookeeper, Oozie, Mahout, Hue, Ambari, Tez, and a real-time version of Hive (Stinger) and other open source tools. Provides Hortonworks high-availability support, a high-performance Hive ODBC driver, and Talend Open Studio for Big Data.
MapR	Based on Hadoop 1, this distribution (Version M7 as of this writing) includes HDFS, HBase, MapReduce, Hive, Mahout, Oozie, Pig, ZooKeeper, Hue, and other open source tools. It also includes direct NFS access, snapshots, and mirroring for "high availability," a proprietary HBase implementation that is fully compatible with Apache APIs, and a MapR management console.
IBM InfoSphere BigInsights	As of this writing, this is based on Hadoop 1 and available in two editions. The Basic Edition includes HDFS, Hbase, MapReduce, Hive, Mahout, Oozie, Pig, ZooKeeper, Hue, and several other open source tools, as well as a basic version of the IBM installer and data access tools. The Enterprise Edition adds sophisticated job management tools, a data access layer that integrates with major data sources, and BigSheets (a spreadsheet-like interface for manipulating data in the cluster).
GreenPlum's Pivotal HD	As of this writing, this is based on Hadoop 2, and includes HDFS, MapReduce, Hive, Pig, HBase, Zookeeper, Sqoop, Flume, and other open source tools. The proprietary advanced Database Services (ADS) powered by HAWQ extends Pivotal HD Enterprise, adding rich, proven, parallel SQL processing facilities.
Amazon Elastic MapReduce (EMR)	As of this writing, this is based on Hadoop 1. Amazon EMR is a web service that enables users to easily and cost-effectively process vast amounts of data. It utilizes a hosted Hadoop framework running on the web-scale infrastructure of Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3). It includes HDFS (with S3 support), HBase (proprietary backup recovery), MapReduce, Hive (added support for Dynamo), Pig, and Zookeeper.
Windows Azure HDInsight	Based on the Hortonworks Data Platform (Hadoop 1), this runs in the Azure cloud. It is integrated with the Microsoft management console for easy deployment and integration with System Center. It can be integrated with Excel through a Hive Excel plug-in. It can be integrated with Microsoft SQL Server Analysis Services (SSAS), PowerPivot, and Power View through the Hive Open Database Connectivity (ODBC) driver. The Azure Marketplace empowers customers to connect to data, smart mining algorithms, and people outside of the users' firewalls. Windows Azure Marketplace offers hundreds of data sets from trusted third-party providers.

Certainly, the abundance of distributions may leave you wondering, "What distribution should I use?" When deciding on a specific distribution for a company/department, you should consider the following:

- **Technical details** — This should encompass, for example, the Hadoop version, included components, proprietary functional components, and so on.
- **Ease of deployment** — This would be the availability of toolkits to manage deployment, version upgrades, patches, and so on.
- **Ease of maintenance** — This would be cluster management, multi-centers support, disaster-recovery support, and so on.
- **Cost** — This would include the cost of implementation for a particular distribution, the billing model, and licenses.
- **Enterprise integration support** — This would include support for integration of Hadoop applications with the rest of the enterprise.

Your choice of a particular distribution depends on a specific set of problems that you are planning to solve by using Hadoop. The discussions in this book are intended to be distribution-agnostic because the authors realize that each distribution provides value.

DEVELOPING ENTERPRISE APPLICATIONS WITH HADOOP

Meeting the challenges brought on by Big Data requires rethinking the way you build applications for data analytics. Traditional approaches for building applications that are based on storing data in the database typically will not work for Big Data processing. This is because of the following reasons:

- The foundation of traditional applications is based on transactional database access, which is not supported by Hadoop.
- With the amount of data stored in Hadoop, real-time access is feasible only on a partial data stored on the cluster.
- The massive data storage capabilities of Hadoop enable you to store versions of data sets, as opposed to the traditional approach of overwriting data.

As a result, a typical Hadoop-based enterprise application will look similar to the one shown in Figure 1-2. Within such applications, there is a *data storage layer*, a *data processing layer*, a *real-time access layer*, and a *security layer*. Implementation of such an architecture requires understanding not only the APIs for the Hadoop components involved, but also their capabilities and limitations, and the role each component plays in the overall architecture.

As shown in Figure 1-2, the data storage layer is comprised of two partitions of source data and intermediate data. *Source data* is data that can be populated from external data sources, including enterprise applications, external databases, execution logs, and other data sources. *Intermediate data* results from Hadoop execution. It can be used by Hadoop real-time applications, and delivered to other applications and end users.

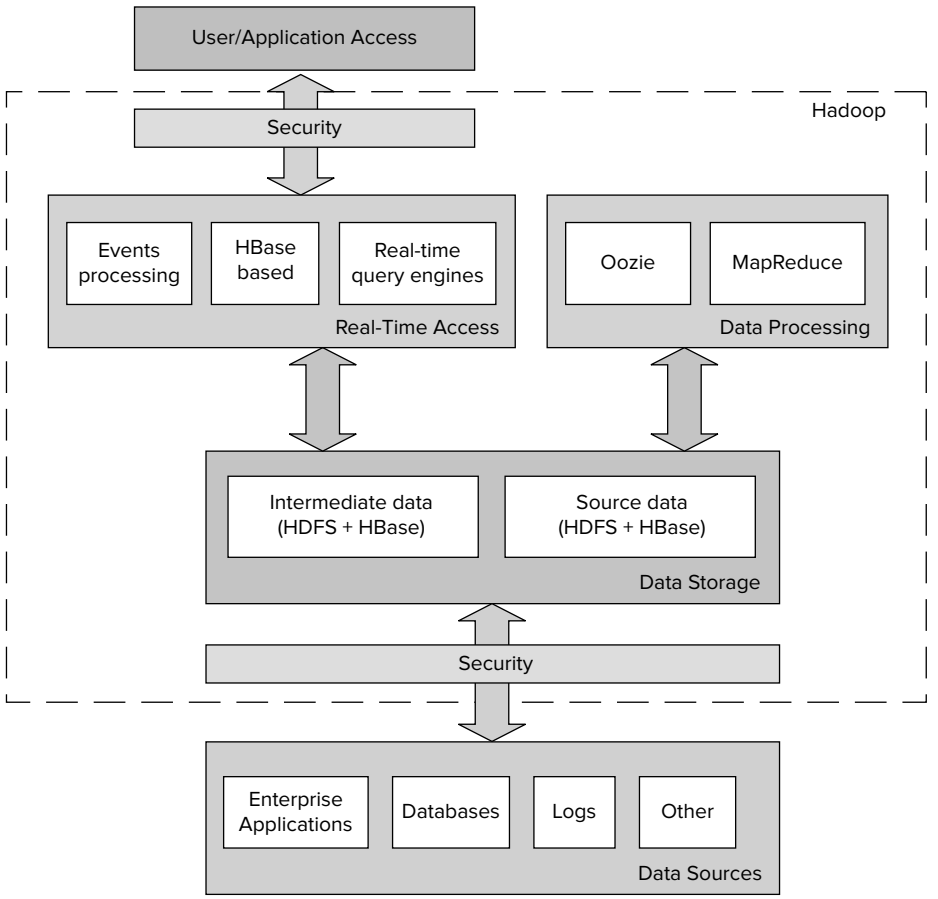

FIGURE 1-2: Notional Hadoop enterprise application

Source data can be transferred to Hadoop using different mechanisms, including Sqoop, Flume, direct mounting of HDFS as a Network File System (NFS), and Hadoop real-time services and applications. In HDFS, new data does not overwrite existing data, but creates a new version of the data. This is important to know because HDFS is a "write-once" filesystem.

For the data processing layer, Oozie is used to combine MapReduce jobs to preprocess source data and convert it to the intermediate data. Unlike the source data, intermediate data is not versioned, but rather overwritten, so there is only a limited amount of intermediate data.

For the real-time access layer, Hadoop real-time applications support both direct access to the data, and execution based on data sets. These applications can be used for reading Hadoop-based intermediate data and storing source data in Hadoop. The applications can also be used for serving users and integration of Hadoop with the rest of the enterprise.

Because of a clean separation of source data used for storage and initial processing, and intermediate data used for delivery and integration, this architecture allows developers to build

applications of virtually any complexity without any transactional requirements. It also makes real-time data access feasible by significantly reducing the amount of served data through intermediate preprocessing.

HADOOP EXTENSIBILITY

Although many publications emphasize the fact that Hadoop hides infrastructure complexity from business developers, you should understand that Hadoop extensibility is not publicized enough.

Hadoop's implementation was designed in a way that enables developers to easily and seamlessly incorporate new functionality into Hadoop's execution. By providing the capability to explicitly specify classes responsible for different phases of the MapReduce execution, Hadoop allows developers to adapt its execution to a specific problem's requirement, thus ensuring that every job is executed in the most cost-effective and performant way.

The main customization of Hadoop execution can include the following:

- Customizing the ways Hadoop parallelizes problem execution, including the way execution is partitioned and the location of execution
- Support for new input data types and locations
- Support for new output data types
- Customizing the locations of the output data

A significant portion of this book is dedicated to describing approaches to such customizations, as well as practical implementations. These are all based on the results of work performed by the authors.

As you will discover, this book covers all the major layers of Hadoop-based enterprise applications shown in Figure 1-2.

Chapter 2 describes the approaches for building a data layer. There you learn about options for building the data layer, including HDFS and HBase (both architecture and APIs). You will also see some comparative analysis of both, along with some guidelines on how to choose one over the other, or use a combination of both. The chapter also covers Avro — Hadoop's new serialization/marshaling framework — and the role it plays in storing or accessing data. Finally, you learn about HCatalog, and the way it can be used to advertise and access data.

The description of data processing represents the bulk of the discussions throughout this book. For the data-processing portion of applications, you will find that the authors recommend using MapReduce and Oozie.

WHY SUCH A FOCUS ON MAPREDUCE CODE IN THIS BOOK?

You may be asking yourself why so much focus is being placed on MapReduce code throughout this book — versus the use of higher-level languages that can make MapReduce programming simpler. You can find a lot of similar discussions on the web and within the Hadoop community about this subject. An argument given in such discussions is that MapReduce code is typically larger (in terms of lines of code) compared to Pig code providing the same functionality. Although this may be an undisputed fact, you have several additional things to consider:

- Not everything can be expressed in a high-level language. Certain things actually require good old-fashioned Java code for their implementation.
- If you are writing a piece of code to be executed once, the number of lines of code may be important to you. But if you are writing enterprise applications, you should consider other criteria, including performance, reliability, and security. Typically, these qualities are easier to implement using MapReduce code, which provides significantly more options.
- By supporting customization of execution, MapReduce provides users with the capability to further improve performance, reliability, and security of applications.

In Chapter 3, you learn about the MapReduce architecture, its main components, and its programming model. The chapter also covers MapReduce application design, some design patterns, and general MapReduce "dos" and "don'ts." The chapter also describes how MapReduce execution is actually happening. As mentioned, one of the strongest MapReduce features is its capability to customize execution. Chapter 4 covers details of customization options and contains a wealth of real-life examples. Chapter 5 rounds out the MapReduce discussion by demonstrating approaches that can help you build reliable MapReduce applications.

Despite the power of MapReduce itself, practical solutions typically require bringing multiple MapReduce applications together, which involves quite a bit of complexity. Integration of MapReduce applications can be significantly simplified by using Hadoop's Workflow/Coordinator engine.

THE VALUE OF OOZIE

One of the most undervalued components of Hadoop is Oozie. Few (if any) of the available books on Hadoop discuss this extremely valuable component. This book demonstrates not only what Oozie can do, but also provides an end-to-end example that shows you how to leverage Oozie functionality to solve practical problems.

continues

continued

Similar to the rest of Hadoop, Oozie functionality is very extensible. You learn about different approaches to extending Oozie functionality.

Finally, one of the most underappreciated Hadoop challenges is integration of Hadoop execution with the rest of the enterprise processing. By using Oozie to orchestrate MapReduce applications and expose these Hadoop processes using Oozie APIs, you have available to you a very elegant integration approach between Hadoop processing and the rest of the enterprise processing.

Chapter 6 describes what Oozie is, its architecture, main components, programming languages, and an overall Oozie execution model. To better explain the capabilities and roles of each Oozie component, Chapter 7 presents end-to-end, real-world applications using Oozie. Chapter 8 completes the Oozie description by showing some of the advanced Oozie features, including custom Workflow activities, dynamic Workflow generation, and uber-`jar` support.

The real-time access layer is covered in Chapter 9. That chapter begins by giving examples of real-time Hadoop applications used by the industry, and then presents the overall architectural requirements for such implementations. You then learn about three main approaches to building such implementations — HBase-based applications, real-time queries, and stream-based processing. The chapter covers the overall architecture and provides two complete examples of HBase-based real-time applications. It then describes a real-time query architecture, and discusses two concrete implementations — Apache Drill and Cloudera's Impala. You will also find a comparison of real-time queries to MapReduce. Finally, you learn about Hadoop-based complex event processing, and two concrete implementations — Storm and HFlame.

Developing enterprise applications requires much planning and strategy related to information security. Chapter 10 focuses on Hadoop's security model.

With the advances of cloud computing, many organizations are tempted to run their Hadoop implementations on the cloud. Chapter 11 focuses on running Hadoop applications in Amazon's cloud using the EMR implementation, and discusses additional AWS services (such as S3), which you can use to supplement Hadoop's functionality. It covers different approaches to running Hadoop in the cloud and discusses trade-offs and best practices.

In addition to securing Hadoop itself, Hadoop implementations often integrate with other enterprise components — data is often imported and exported to and from Hadoop. Chapter 12 focuses on how enterprise applications that use Hadoop are best secured, and provides examples and best practices of securing overall enterprise applications leveraging Hadoop.

SUMMARY

This chapter has provided a high-level overview of the relationship between Big Data and Hadoop. You learned about Big Data, its value, and the challenges it creates for enterprises, including data storage and processing. You were also introduced to Hadoop, and learned about a bit of its history.

You were introduced to Hadoop's features, and learned why Hadoop is so well-suited for Big Data processing. This chapter also showed you an overview of the main components of Hadoop, and presented examples of how Hadoop can simplify both data science and the creation of enterprise applications.

You learned a bit about the major Hadoop distributions, and why many organizations tend to choose particular vendor distributions because they do not want to deal with the compatibility of individual Apache projects, or might need vendor support.

Finally, this chapter discussed a layered approach and model for developing Hadoop-based enterprise applications.

Chapter 2 starts diving into the details of using Hadoop, and how to store your data.

CONTENTS