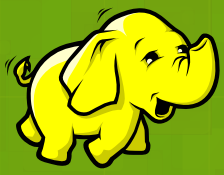
The Executive's Guide To

BIG DATA & APACHE HADOOP



Everything you need to understand and get started with Big Data and Hadoop

Robert D. Schneider
Author of Hadoop for Dummies

The Executive's Guide To

BIG DATA & APACHE HADOOP

Introduction 4
Introducing Big Data 4
What Turns Plain Old Data into Big Data? 5
Larger Amounts of Information 5
Comparing Database Sizes 6
More Types of Data 8
Relational 10
Columnar 10
Key/Value 10
Documents, Files, and Objects 10
Graph 10
Generated by More Sources 11
Retained for Longer Periods 12
Utilized by More Types of Applications 12
Implications of Not Handling Big Data Properly 13
Checklist: How to Tell When Big Data Has Arrived 14
Distributed Processing Methodologies 15
Hadoop 18
Checklist: Ten Things to Look for When Evaluating Hadoop Technology 24
Hadoop Distribution Comparison Chart 27
Glossary of Terms 28
About the Author 32

Introduction

It seems that everywhere you look – in both the mainstream press as well as in technology media – you see stories or news reports extolling Big Data and its revolutionary potential. But dig a little deeper, and you'll discover that there's great confusion about Big Data in terms of exactly what it is, how to work with it, and how you can use it to improve your business.

In this book, I introduce you to Big Data, describing what it consists of and what's driving its remarkable momentum. I also explain how distributed, parallel processing methodologies – brought to life in technologies such as Hadoop and its thriving ecosystem – can help harvest knowledge from the enormous volumes of raw data – both structured and unstructured – that so many enterprises are generating today. In addition, I point out that this is a highly dynamic field, with nonstop innovation that goes far beyond the original batch processing scenarios to innovative new use cases like streaming, real-time analysis, and pairing machine learning with SQL.

Finally, I provide some benchmarks that you can use to confirm that Big Data has indeed arrived in your organization, along with some suggestions about how to proceed.

The intended audience for this book includes executives, IT leaders, line-of-business managers, and business analysts.

Introducing Big Data

Big Data has the potential to transform the way you run your organization. When used properly it will create new insights and more effective ways of doing business, such as:

- How you design and deliver your products to the market
- How your customers find and interact with you
- Your competitive strengths and weaknesses
- Procedures you can put to work to boost the bottom line

What's even more compelling is that if you have the right technology infrastructure in place, many of these insights can be delivered in real-time. Furthermore, this newfound knowledge isn't just academic: you can apply what you learn to improve daily operations.

What Turns Plain Old Data into Big Data?

It can be difficult to determine when you've crossed the nebulous border between normal data operations and the realm of Big Data. This is particularly tough since Big Data is often in the eye of the beholder. Ask ten people about what Big Data is, and you'll get ten different answers.

From my perspective, organizations that are actively working with Big Data have each of the following five traits in comparison to those who don't:

1. Larger amounts of information
2. More types of data
3. Data that's generated by more sources
4. Data that's retained for longer periods
5. Data that's utilized by more types of applications

Let's examine the implications of each of these Big Data properties.

Larger Amounts of Information

Thanks to existing applications, as well as new sources that I'll soon describe, enterprises are capturing, storing, managing, and using more data than ever before. Generally, these events aren't confined to a single organization; they're happening everywhere:

- On average over 500 million Tweets occur every day
- World-wide there are over 1.1 million credit card transactions every second
- There are almost 40,000 ad auctions per second on Google AdWords
- On average 4.5 billion "likes" occur on Facebook every day

Let's take a look at the differences between common sizes of databases.

Comparing Database Sizes

It's easy to fall into the trap of flippantly tossing around terms like gigabytes, terabytes, and petabytes without truly considering the truly impressive differences in scale among these vastly different volumes of information. Table 1 below summarizes the traits of a **1-gigabyte**, **1-terabyte**, and **1-petabyte** database.

Database Size	Common Characteristics
1 gigabyte	····> Information generated by traditional enterprise applications ····> Typically consists of transactional data, stored in relational databases ····> Uses Structured Query Language (SQL) as the access method
1 terabyte	····> Standard size for data warehouses ····> Often aggregated from multiple databases in the 1-100 gigabyte range ····> Drives enterprise analytics and business intelligence
1 petabyte	····> Frequently populated by mass data collection – often automated ····> Regularly contains unstructured information ····> Serves as a catalyst for exploring new Big Data-related technologies

Table 1: *Representative Characteristics for Today's Databases*

Figure 1 Compares the relative scale of a 1-gigabyte, 1-terabyte, and 1-petabyte database.

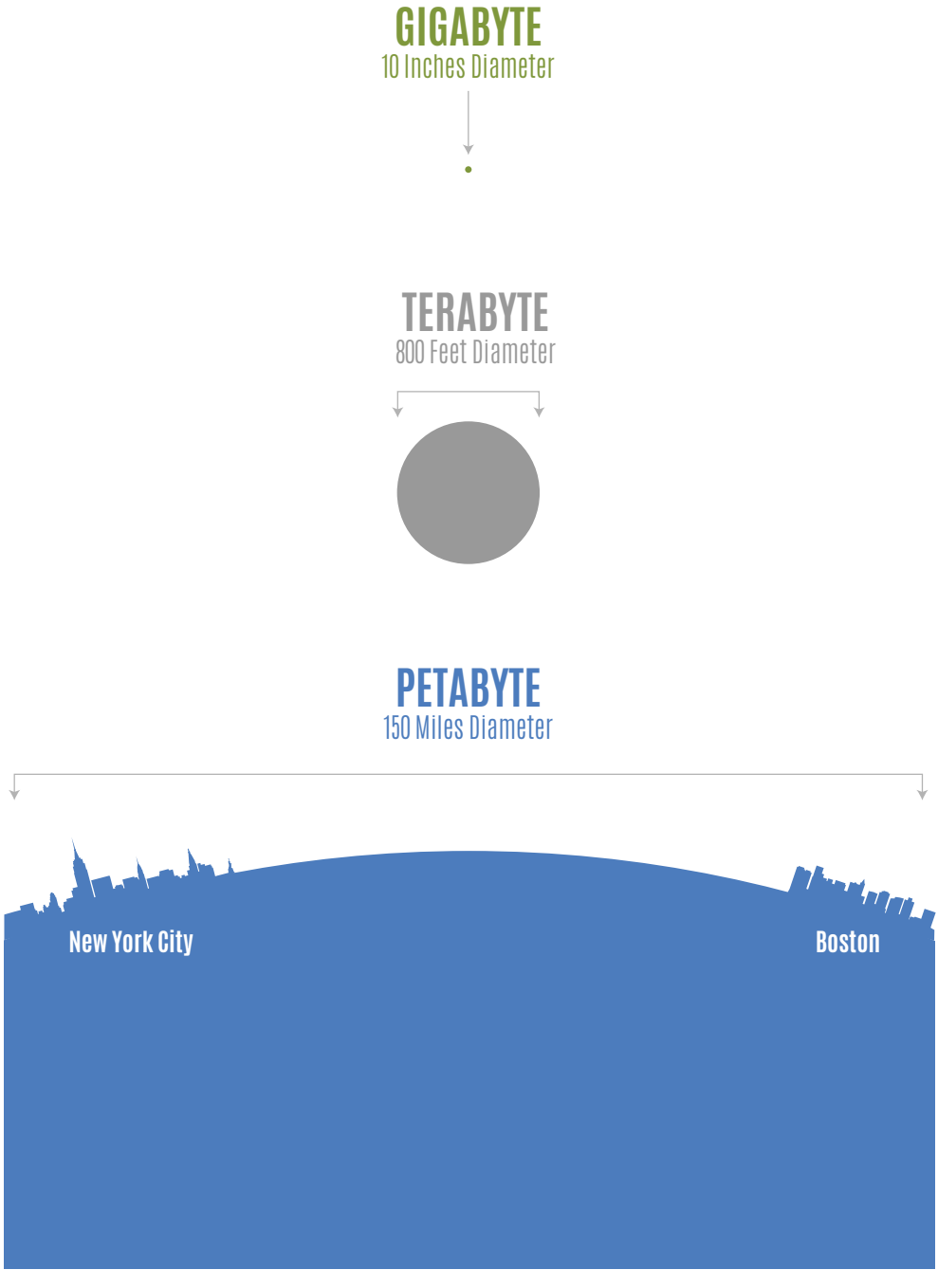


Figure 1: *Relative Scale of Databases*

More Types of Data

Structured data – regularly generated by enterprise applications and amassed in relational databases – is usually clearly defined and straightforward to work with. On the other hand, enterprises are now interacting with enormous amounts of unstructured – or semi-structured – information, such as:

- Clickstreams and logs from websites
- Photos
- Video
- Audio
- XML documents
- Freeform blocks of text such as email messages, Tweets, and product reviews

Figure 2 illustrates the types of unstructured and structured data.

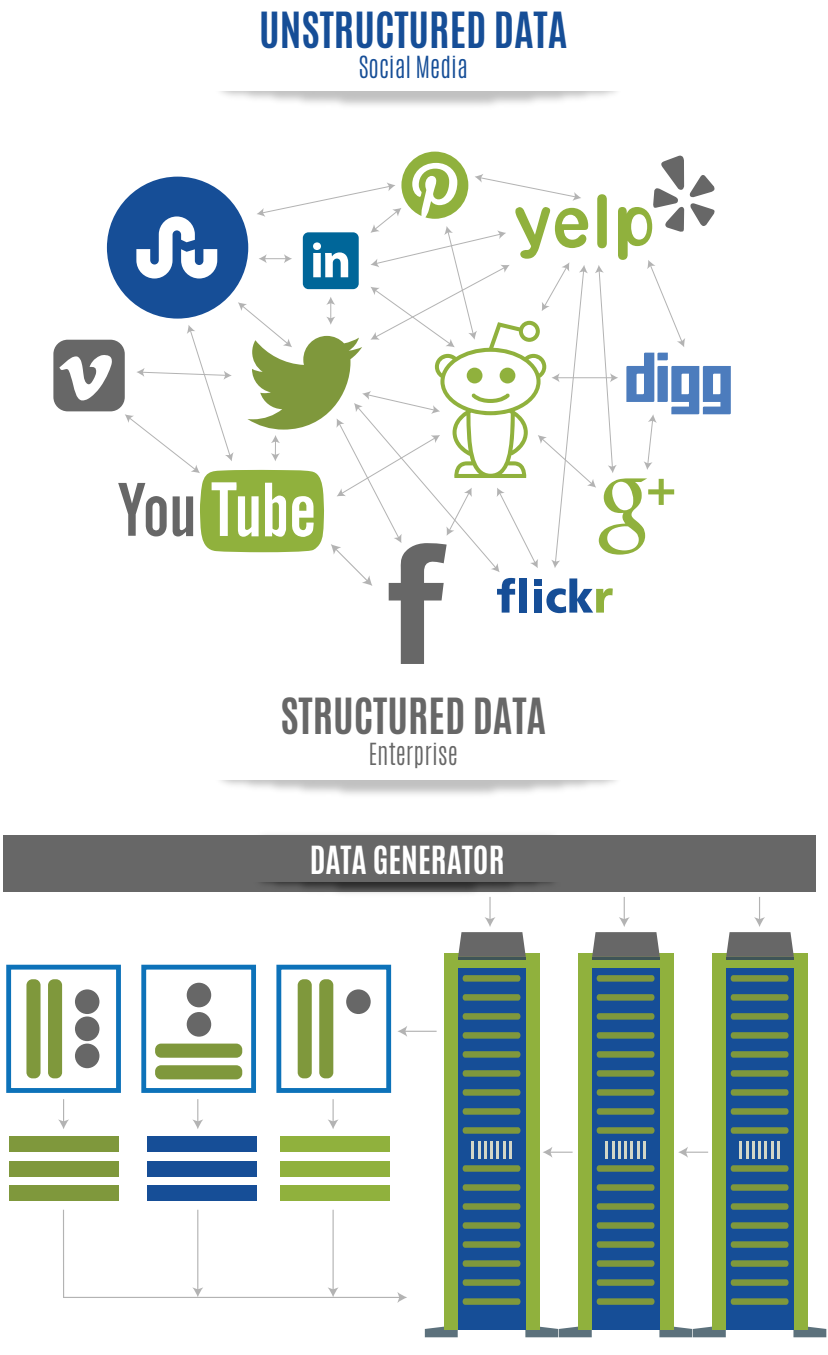


Figure 2: *Unstructured Data vs Structured Data*

Prior to the era of Big Data, mainstream information management solutions were fairly straightforward, and primarily consisted of relational databases. Today, thanks to the widespread adoption of Big Data, the average IT organization must provide and support many more information management platforms. It's also important to remember that to derive the maximum benefits from Big Data, you must take all of your enterprise's information into account.

Below are details about some of the most common data technologies found in today's Big Data environments.

Relational

Dating back to late 1970s, relational databases (RDBMS) have had unprecedented success and longevity. Their information is usually generated by transactional applications, and these databases continue to serve as the preferred choice for storing critical corporate data. Relational databases will continue to remain an integral player in Big Data environments, because:

- SQL and set-based processing have been wildly successful
- The relations among data are essential to the enterprise
- Transactional integrity (i.e. ACID compliance) is critical
- There's an enormous installed base of applications and developer/administrative talent

Columnar

Just like their relational database siblings, columnar databases commonly hold well-structured information. However, columnar databases persist their physical information on disk by columns, rather than by rows. This yields big performance increases for analytics and business intelligence.

Key/Value

Also known as field-value, name-value, or attribute-value pairs, these are customarily used for capturing, storing, and querying fine-grained name/value pairs. This includes data from device monitoring, timestamps, metadata, and so on.

Key/value as a concept goes back over 30 years, but it's really come into its own with the rise of massive web logs.

Documents, Files, and Objects

Object-oriented data has traditionally been difficult to store in an RDBMS. Additionally, organizations are now capturing huge volumes of binary data such as videos, images, and document scans. These are frequently placed in specialized data repositories that are customized for this type of information.

Graph

Graph databases are meant to express relationships among a limitless set of elements. They let users and applications traverse these connections very quickly and get the answers to some very complex queries. They are exceptionally powerful when combined with relational information, answering questions like "what did a particular person's friends buy from our website?" or "which of our employees have a family member that is working for a large customer?" These databases form the foundation of social networks such as Facebook and LinkedIn.

Each of these five, highly specialized database technologies does a great job of working with its own respective information categories. Unfortunately, numerous IT organizations are discovering that these platforms are having difficulties keeping pace with the relentless influx of data – especially unstructured – and are extraordinarily expensive to scale.

Generated by More Sources

Enterprise applications continue to produce transactional and web data, but there are many new conduits for generating information, including:

- Smartphones
- Medical devices
- Sensors

- GPS location data
- Machine-to-machine, streaming communication

Retained for Longer Periods

Government regulations, industry standards, company policies, and user expectations are all contributing to enterprises keeping their data for lengthier amounts of time. Many IT leaders also recognize that there are likely to be future use cases that will be able to profit from historical information, so carelessly throwing data away isn't a sound business strategy. However, hoarding vast and continually growing amounts of information in core application storage is prohibitively expensive. Instead, migrating information to Hadoop is significantly less costly, plus Hadoop is capable of handling a much bigger variety of data.

Utilized by More Types of Applications

Faced with a flood of new information, many enterprises are following a "grab the data first, and then figure out what to do with it later" approach. This means that there are countless new applications being developed to work with all of this diverse information. Such new applications are widely varied, yet must satisfy requirements such as bigger transaction loads, faster speeds, and enormous workload variability.

Big Data is also shaking up the analytics landscape. Structured data analysis has historically been the prime player, since it works well with traditional relational database-hosted information. However, driven by Big Data, unstructured information analysis is quickly becoming equally important. Several new techniques work with data from manifold sources such as:

- Blogs
- Facebook
- Twitter
- Web traffic logs
- Text messages
- Yelp reviews

- Support desk calls
- Call center calls

By itself, Big Data is interesting. But things really get intriguing when you blend it with traditional sources of information to come up with innovative solutions that produce significant business value. For example, a manufacturer could tie together its inventory availability - contained in a relational database - with images and video instructions from a document store-based product catalog. The resulting solution would help customers to immediately select and order the correct part. In another scenario, an e-commerce vendor might meld a given customer's purchase history from a relational database with what other clients with similar profiles have been buying, details that would be retrieved from a graph database. This could power a very accurate recommendation engine for presenting new products. Finally, a hotel might join property search results from a key/value database with historical occupancy metrics in a relational database to optimize nightly pricing and consequently achieve better yield management.

Implications of Not Handling Big Data Properly

Failing to keep pace with the immense data volumes, mushrooming number of information sources and categories, longer data retention periods, and expanding suite of data-hungry applications has impeded many Big Data plans, and is resulting in:

- Delayed or faulty insights
- An inability to detect and manage risk
- Diminished revenue
- Increased cost
- Opportunity costs of missing new applications along with operational use of data
- A weakened competitive position

Fortunately, new tools and technologies are arriving to help make sense of Big Data; distributed processing methodologies and Hadoop are prime examples of fresh thinking to address Big Data.

Checklist: How to Tell When Big Data Has Arrived

1. *You're getting overwhelmed with raw data from mobile or medical devices, sensors, and/or machine-to-machine communications. Additionally, it's likely that you're so busy simply capturing this data that you haven't yet found a good use for it.*

2. *You belatedly discover that people are having conversations about your company on Twitter. Sadly, not all of this dialogue is positive.*

3. *You're keeping track of a lot more valued information from many more sources, for longer periods of time. You realize that maintaining such extensive amounts of historical data might present new opportunities for deeper awareness into your business.*

4. *You have lots of silos of data, but can't figure out how to use them together. You may already be deriving some advantages from limited, standalone analysis, but you know that the whole is greater than the sum of the parts.*

5. *Your internal users – such as data analysts – are clamoring for new solutions to interact with all this data. They may already be using one-off analysis tools such as spreadsheets, but these ad-hoc approaches don't go nearly far enough.*

6. *Your organization seeks to make real-time business decisions based on newly acquired information. These determinations have the potential to significantly impact daily operations.*

7. *You've heard rumors (or read articles) about how your competitors are using Big Data to gain an edge, and you fear being left behind.*

8. *You're buying lots of additional storage each year. These supplementary resources are expensive, yet you're not putting all of this extra data to work.*

9. *You've implemented – either willingly or by necessity – new information management technologies, often from startups or other cutting-edge vendors. However, many of these new solutions are operating in isolation from the rest of your IT portfolio.*

Distributed Processing Methodologies

In the past, organizations that wanted to work with large information sets would have needed to:

- Acquire very powerful servers, each sporting very fast processors and lots of memory
- Stage massive amounts of high-end, often-proprietary storage
- License an expensive operating system, a RDBMS, business intelligence, and other software
- Hire highly skilled consultants to make all of this work
- Budget lots of time and money

Since all of the above steps were so complex, pricey, and lengthy, it's no wonder that so many enterprises shied away from undertaking these projects in the first place. In those rare instances where an organization took the plunge, they commonly restricted interaction with the resulting system. This gated access was feasible when the amounts of data in question were measured in gigabytes and the internal user community was rather small.

However, this approach no longer works in a world where data volumes grow by more than 50% each year and are tallied in terabytes – and beyond. Meanwhile, much of this information is unstructured, and increasing numbers of employees are demanding to interact with all of this data.

Fortunately, several distinct but interrelated technology industry trends have made it possible to apply fresh strategies to work with all this information:

- Commodity hardware
- Distributed file systems
- Open source operating systems, databases, and other infrastructure
- Significantly cheaper storage
- Widespread adoption of interoperable Application Programming Interfaces (APIs)

Today, there's an intriguing collection of powerful distributed processing methodologies to help derive value from Big Data.

In a nutshell, these distributed processing methodologies are constructed on the proven foundation of 'Divide and Conquer': it's much faster to break a massive task into smaller chunks and process them in parallel. There's a long history of this style of computing, dating all the way back to functional programming paradigms like LISP in the 1960s.

Given how much information it must manage, Google has long been heavily reliant on these tactics. In 2004, Google published a white paper that described their thinking on parallel processing of large quantities of data, which they labeled "MapReduce". The white paper was conceptual in that it didn't spell out the implementation technologies per se. Google summed up MapReduce as follows:

> *"MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key."*

MapReduce was proven to be one of the most effective techniques for conducting batch-based analytics on the gargantuan amounts of raw data generated by web search and crawling before organizations expanded their use of MapReduce to additional scenarios.

Figure 3 below illustrates the concepts behind MapReduce.

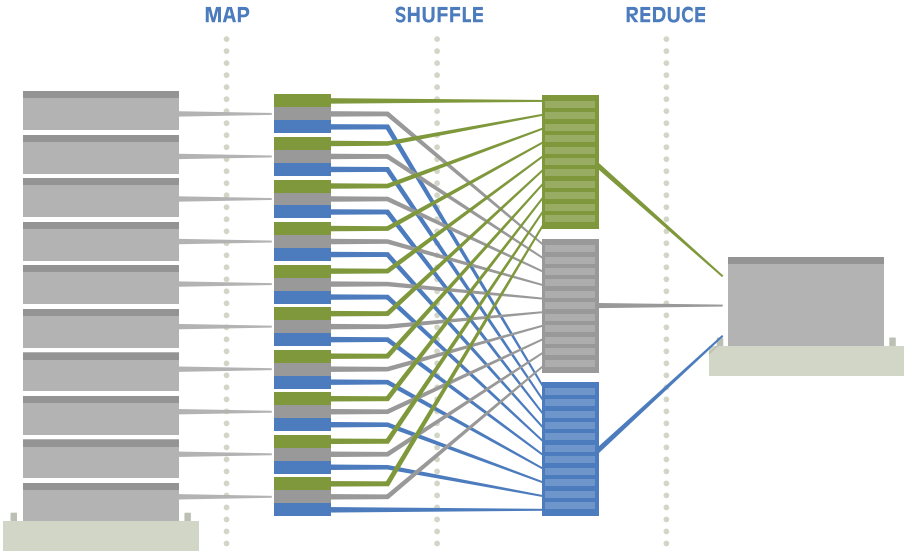


Figure 3: *Map and Reduce Processing*

Rather than referring to a single tactic, MapReduce is actually a collection of complementary processes and strategies that begins by pairing commoditized hardware and software with specialized underlying file systems. Computational tasks are then directly performed on the data wherever it happens to reside, rather than the previous practices of first copying and aggregating raw data into a single repository before processing it. These older practices simply won't scale when the amount of data expands beyond terabytes. Instead, MapReduce's innovative thinking means that rather than laboriously moving huge volumes of raw data across a network, only code is sent over the network.

MapReduce was, and continues to be, a superb strategy for the problem that it was originally designed to solve: how to conduct batch analysis on the massive quantities of data generated by users running searches and visiting web sites. The concepts behind MapReduce have also served as the inspiration for an ever-expanding collection of novel parallel processing computational frameworks aimed at a variety

of use cases, such as streaming analysis, interactive querying, integrating SQL with machine learning, and so on. While not all of these new approaches will achieve the same level of traction as the popular and still-growing batch-oriented MapReduce, many are being used to solve interesting challenges and drive new applications.

Conveniently, each of these methodologies shields software developers from the thorny challenges of distributed, parallel processing. But as I described earlier, Google's MapReduce paper didn't dictate exactly what technologies should be used to implement its architecture. This means that unless you worked for Google, it's unlikely that you had the time, money, or people to design, develop, and maintain your own, site-specific set of all of the necessary components for systems of this sophistication. After all, it's doubtful that you built your own proprietary operating system, relational database management system, or Web server.

Thus, there was a need for a complete, standardized, end-to-end solution suitable for enterprises seeking to apply the full assortment of modern, distributed processing techniques to help extract value from reams of Big Data. This is where Hadoop comes in.

Hadoop

Around the same time that Google was publishing the MapReduce paper, two engineers - Doug Cutting and Mike Cafarella - were busily working on their own web crawling technology named Nutch. After reading Google's research, they quickly adjusted their efforts and set out to create the foundations of what would later be known as Hadoop. Eventually, Cutting joined Yahoo! where the Hadoop technology was expanded further. As Hadoop grew in sophistication, Yahoo! extended its usage into additional internal applications. In early 2008, the Apache Software Foundation (ASF) promoted Hadoop into a top-level open source project.

Simply stated, Hadoop is a comprehensive software platform that executes distributed data processing techniques. It's implemented in several distinct, specialized modules:

- Storage, principally employing the Hadoop File System (HDFS) although other more robust alternatives are available as well
- Resource management and scheduling for computational tasks
- Distributed processing programming model based on MapReduce
- Common utilities and software libraries necessary for the entire Hadoop platform

Hadoop is also at the center of a diverse, flourishing network of ancillary projects and solutions that I will describe later.

Hadoop has broad applicability across all industries. Table 2 shows four distinct usage categories, along with some example applications in each grouping.

Category	Example Applications
Enterprise Data Hub	Ultra-fast data ingestion Multi-structured data staging Extract/transform/load and data warehousing offload Mainframe offload Investigative analytics Simple query and reporting
Market Optimization and Targeting	Cross-channel behavioral analysis Social media analysis Click-stream analysis Recommendation engines and targeting Advertising impression and conversion analysis
Risk Detection and Prevention	Network security monitoring Security information and event management Fraudulent behavioral analysis Bot detection and prevention

Operations Intelligence	····> Supply chain and logistics ····> System log analysis ····> Assembly line quality assurance ····> Preventative maintenance ····> Smart meter analysis

Table 2: *Example Hadoop Applications*

Enterprises have responded enthusiastically to Hadoop. Table 3 below illustrates just a few examples of how Hadoop is being used in production today.

Industry	Real-World Hadoop Applications
Financial Services	This industry offers some very interesting optimization prospects because of the huge amounts of data that it generates, its tight processing windows, strict regulatory and reporting requirements, and the ever-present potential for fraudulent or risky behavior. Hadoop is able to apply distributed processing methodologies that excel in conducting the pattern matching necessary to detect fraud or other nefarious activities. It can incorporate hundreds – or even thousands – of indicators to help improve credit score accuracy while also flagging potential risk situations before they can proceed.
Publishing	Analyze user interactions with mobile reading devices to deliver precise search results as well as more meaningful recommendations. Since these data-driven suggestions are accurate, fine-tuned, and timely, users are more likely to make additional purchases and be satisfied with what they've bought.

Healthcare	It's well known that the job of designing new pharmaceutical products is both costly and very risky. Employing Hadoop for massive data storage and then applying analytics to process and correlate raw financial, patient, and drug data speeds up drug development, improves patient care, and ultimately reduces total healthcare costs across the system.
Retail	Load and then process massive amounts of information – such as website searches, shopping cart interactions, tailored promotion responses, and inventory management – to gain a better understanding of customer buying trends. Rapidly analyzing all of these data points from separate systems makes it possible for the retailer to tailor its prices and promotions based on actual intelligence, rather than hunches.
Advertising	Online advertising systems produce massive amounts of information in the blink of an eye. For example, there are almost 40,000 ad auctions per second on Google AdWords. Even the slightest improvement in advertisement pricing yields tremendous profitability advancements. But these optimizations are only possible if they're conducted in real-time by using Hadoop to analyze conversion rates and the cost to serve ads, and then applying this knowledge to drive incremental revenue.

Table 3: *Real-World Hadoop Applications*

Rather than viewing Hadoop as a single, monolithic solution, it's better to regard it as a platform with an assortment of applications built on its foundation. Over time, Hadoop's success has also spawned a rich ecosystem, as shown in Figure 4 below.

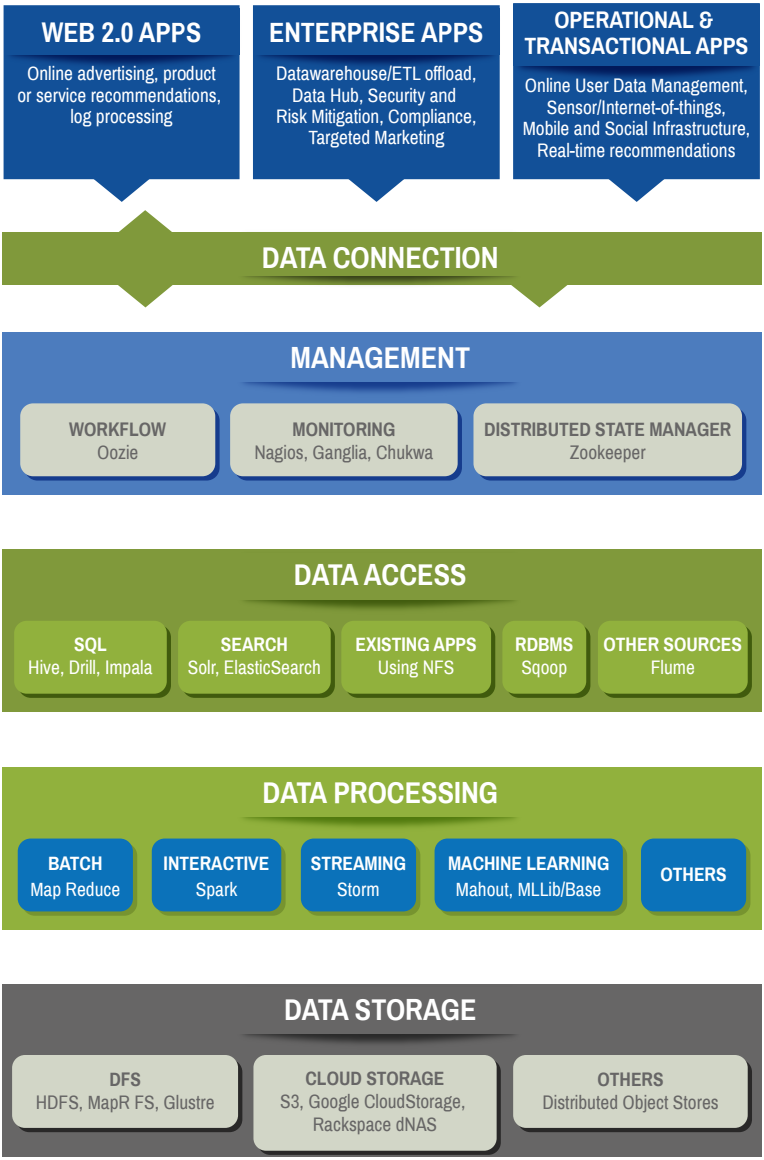
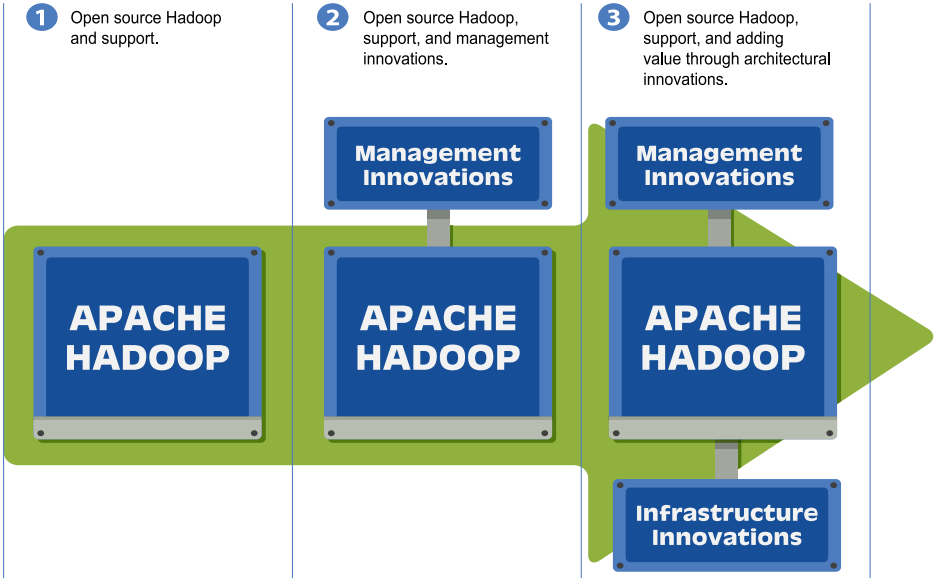


Figure 4: *Hadoop's Ecosystem*

See the glossary for more details about each of the elements in the Hadoop ecosystem.

With all of these moving parts, there are now several distinct options for organizations seeking to deploy Hadoop and its related technologies. These generally fall into one of three implementation models:

1. **Open source Hadoop and support.** This pairs bare-bones open source with paid professional support and services. Hortonworks is a good example of this model.
2. **Open source Hadoop and management utilities.** This goes a step further by joining open source Hadoop with IT-friendly tools and utilities that make things easier for mainline IT organizations. Cloudera is an instance of this model.
3. **Open source Hadoop, management utilities, and innovative added value at all layers – including Hadoop's foundation.** Some vendors are enhancing Hadoop's capabilities with enterprise-grade features yet still remaining faithful to the core open source components. MapR is the best-known adherent to this approach.



Selecting your Hadoop infrastructure is a vital IT decision that will affect the entire organization for years to come, in ways that you can't visualize now. This is particularly true since we're only at the dawn of Big Data in the enterprise. Hadoop is no longer an "esoteric", lab-oriented technology; instead, it's becoming mainline, it's continually evolving, and it must be integrated into your enterprise. Selecting a Hadoop implementation requires the same level of attention and devotion as your organization expends when choosing other critical core technologies, such as application servers, storage, and databases. You can expect your Hadoop environment to be subject to the same requirements as the rest of your IT asset portfolio, including:

- Service Level Agreements (SLAs)
- Data protection
- Security
- Integration with other applications

Checklist: Ten Things to Look for When Evaluating Hadoop Technology

1	***Look for solutions that support open source and ecosystem components that support Hadoop API's. It's wise to make sure API's are open to avoid lock-in.***
2	***Interoperate with existing applications. One way to magnify the potential of your Big Data efforts is to enable your full portfolio of enterprise applications to work with all of the information you're storing in Hadoop.***
3	***Examine the ease of migrating data into and out of Hadoop. By mounting your Hadoop cluster as an NFS volume, applications can load data directly into Hadoop and then gain real-time access to Hadoop's results. This approach also increases usability by supporting multiple concurrent random access readers and writers.***

Checklist: Ten Things to Look for When Evaluating Hadoop Technology

4

Use the same hardware for OLTP and analytics. It's rare for an organization to maintain duplicate hardware and storage environments for different tasks. This requires a high-performance, low-latency solution that doesn't get bogged down with time-consuming tasks such as garbage collection or compactions. Reducing the overhead of the disk footprint and related I/O tasks helps speed things up and increases the likelihood of efficient execution of different types of processes on the same servers.

5

Focus on scalability. In its early days, Hadoop was primarily used for offline analysis. Although this was an important responsibility, instant responses weren't generally viewed as essential. Since Hadoop is now driving many more types of use cases, today's Hadoop workloads are highly variable. This means that your platform must be capable of gracefully and transparently allocating additional resources on an as-needed basis without imposing excessive administrative and operational burdens.

6

Ability to provide real-time insights on newly loaded data. Hadoop's original use case was to crawl and index the Web. But today – when properly implemented – Hadoop can deliver instantaneous understanding of live data, but only if fresh information is immediately available for analysis.

7

A completely integrated solution. Your database architects, operations staff, and developers should focus on their primary tasks, instead of trying to install, configure, and maintain all of the components in the Hadoop ecosystem.

8

Safeguard data via multiple techniques. Your Hadoop platform should facilitate duplicating both data and metadata across multiple servers using practices such as replication and mirroring. In the event of an outage on a particular node you should be able to immediately recover data from where it has been replicated in the cluster. This not only fosters business continuity, it also presents the option of offering read-only access to information that's been replicated to other nodes. Snapshots - which should be available for both files and tables - provide point-in-time recovery capabilities in the event of a user or application error.

Checklist: Ten Things to Look for When Evaluating Hadoop Technology

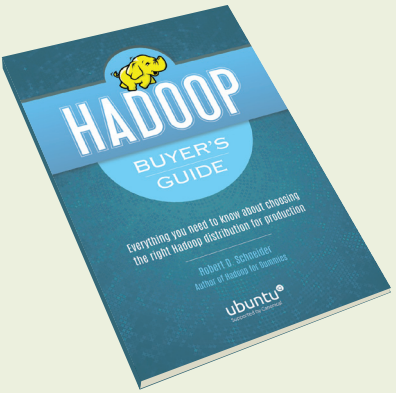
9 *Offer high availability. Hadoop is now a critical enterprise technology infrastructure. Like other enterprise-wide fundamental software assets, it should be possible to upgrade your Hadoop environment without shutting it down. Furthermore, your core Hadoop system should be isolated from user tasks so that runaway jobs can't degrade or even bring down your entire cluster.*

10 *Complete administrative tooling and comprehensive security. It should be easy for your operational staff to maintain your Hadoop landscape, with minimal amounts of manual procedures. Self-tuning is an excellent way that a given Hadoop environment can reduce administrative overhead, and it should also be easy for you to incorporate your existing security infrastructure into Hadoop.*

If you're ready to take the next step on the road to Hadoop, I recommend that you read the Hadoop Buyer's Guide and use the following comparison chart from my other book to help you make the best decision for your organization.

You can download the
Hadoop Buyer's Guide

www.HadoopBuyersGuide.com



One the next page is a quick comparison chart of some of the differences across the major Hadoop distributions.

	Hortonworks	Cloudera	MapR
Performance and Scalability			
Data Ingest	Batch	Batch	Batch and streaming writes
Metadata Architecture	Centralized	Centralized	Distributed
HBase Performance	Latency spikes	Latency spikes	Consistent low latency
NoSQL Applications	Mainly batch applications	Mainly batch applications	Batch and online/real-time applications
Dependability			
High Availability	Single failure recovery	Single failure recovery	Self healing across multiple failures
MapReduce HA	Restart jobs	Restart jobs	Continuous without restart
Upgrading	Planned downtime	Rolling upgrades	Rolling upgrades
Replication	Data	Data	Data + metadata
Snapshots	Consistent only for closed files	Consistent only for closed files	Point-in-time consistency for all files and tables
Disaster Recovery	No	File copy scheduling (BDR)	Mirroring
Manageability			
Management Tools	Ambari	Cloudera Manager	MapR Control System
Volume Support	No	No	Yes
Heat map, Alarms, Alerts	Yes	Yes	Yes
Integration with REST API	Yes	Yes	Yes
Data and Job Placement Control	No	No	Yes
Data Access			
File System Access	HDFS, read-only NFS	HDFS, read-only NFS	HDFS, read/write NFS (POSIX)
File I/O	Append only	Append only	Read/write
Security: ACLs	Yes	Yes	Yes
Wire-level Authentication	Kerberos	Kerberos	Kerberos, Native

Glossary of Terms
(Big Data Concepts, Hadoop, and its Ecosystem)

As is the case with any new technology wave, learning about Big Data - and its supporting infrastructure - means getting comfortable with many new concepts and terms. In this section, I provide a listing - and basic explanation - of some of the most common vocabulary you're likely to encounter as you explore Big Data and Hadoop.

Before you start reviewing these definitions, remember the relationships among Big Data, distributed processing methodologies, and Hadoop:

1. **Big Data.** This is the reality that most enterprises face regarding coping with lots of new information, arriving in many different forms, and with the potential to provide insights that can transform the business.
2. **Distributed processing methodologies**. These procedures leverage the power of multiple computers to divide and conquer even the biggest data collections by breaking large tasks into small, then assigning work to individual computers, and finally reassembling the results to answer important questions. MapReduce is a prominent example of a distributed processing methodology, with many other offshoots also enjoying success, including streaming, real-time analysis, and machine learning.
3. **Hadoop.** A comprehensive technology offering that employs distributed processing methodologies to make the most of Big Data. Hadoop is at the center of a thriving ecosystem of open source solutions and value-added products.

- **Apache Software Foundation.** *A non-profit corporation that manages numerous collaborative, consensus-based open source projects, including the core technologies that underlay and interact with MapReduce and Hadoop.*
- **Avro.** *Serialization and remote procedure capabilities for interacting with Hadoop, using the JSON data format. Offers a straightforward approach for portraying complex data structures within a Hadoop MapReduce job. (Apache Software Foundation project)*
- **Big Data.** *This is the reality that most enterprises face regarding coping with lots of new data, arriving in many different forms, and with the potential to provide insights that can transform the business.*

- **Big Table.** *High performance data storage technology developed at Google, but not distributed elsewhere. Served as an inspiration for Apache HBase.*
- **Cascading.** *Abstraction layer meant to exploit the power of Hadoop while simplifying the job of designing and building data processing operations. This means that developers don't need to learn how to program in MapReduce; they can use more familiar languages such as Java.*
- **Cluster.** *Large-scale Hadoop environment commonly deployed on a collection of inexpensive, commodity servers. Clusters achieve high degrees of scalability merely by adding extra servers when needed, and frequently employ replication to increase resistance to failure.*
- **Data Processing: batch.** *Analyzing or summarizing very large quantities of information with little to no user interaction while the task is running. Results are then presented to the user upon completion of the processing.*
- **Data Processing: interactive.** *Live user-driven interactions with data (through query tools or enterprise applications) that produce instantaneous results.*
- **Data Processing: real-time.** *Machine-driven interactions with data - often continuous. The results of this type of processing commonly serve as input to subsequent real-time operations.*
- **DataNode.** *Responsible for storing data in the Hadoop File System. Data is typically replicated across multiple DataNodes to provide redundancy.*
- **Drill.** *Open source framework targeted at exploiting the power of parallel processing to facilitate high-speed, real-time interactions – including ad-hoc analysis – with large data sets. (Apache Software Foundation project)*
- **Extensible Markup Language (XML).** *A very popular way of representing unstructured/semi-structured information. Text-based and human-readable, there are now hundreds of different XML document formats in use.*
- **Flume.** *Scalable technology developed at Facebook, commonly used to capture log information and write it into the Hadoop File System. (Apache Software Foundation project)*
- **GitHub.** *Internet-based hosting service for managing the software development and delivery process, including version control.*
- **Hadoop.** *A specific approach for implementing the MapReduce architecture, including a foundational platform and a related ecosystem. (Apache Software Foundation project)*

- **Hadoop File System (HDFS).** *File system designed for portability, scalability, and large-scale distribution. Written in Java, HDFS employs replication to help increase reliability of its storage. However, HDFS is not POSIX-compliant. (Apache Software Foundation project)*
- **HBase.** *A distributed – but non relational – database that runs on top of the Hadoop File System. (Apache Software Foundation project)*
- **Hive.** *Data warehousing infrastructure constructed on top of Hadoop. Offers query, analysis, and data summarization capabilities. (Apache Software Foundation project)*
- **Impala.** *A query engine that works with Hadoop and offers SQL language searches on data stored in the Hadoop File System and HBase database.*
- **JavaScript Object Notation (JSON).** *An open data format standard. Language independent, and human-readable, often used as a more efficient alternative to XML.*
- **Machine Learning.** *An array of techniques that evaluate large quantities of information and derive automated insights. After a sufficient number of processing cycles, the underlying algorithms become more accurate and deliver better results - all without human intervention.*
- **Mahout.** *A collection of algorithms for classification, collaborative filtering, and clustering that deliver machine learning capabilities. Commonly implemented on top of Hadoop. (Apache Software Foundation project)*
- **MapReduce.** *Distributed, parallel processing techniques for quickly deriving insight into often-massive amounts of information.*
- **Maven.** *A tool that standardizes and streamlines the process of building software, including managing dependencies among external libraries, components, and packages. (Apache Software Foundation project)*
- **Mirroring.** *A technique for safeguarding information by copying it across multiple disks. The disk drive, operating system, or other specialized software can provide mirroring.*
- **NameNode.** *Maintains directory details of all files in the Hadoop File System. Clients interact with the NameNode whenever seek to locate or interact with a given file. The NameNode responds to these inquiries by returning a list of the DataNode servers where the file in question resides.*
- **Network file system (NFS).** *A file system protocol that makes it possible for both end users and processes on one computer to transparently access and interact with data stored on a remote computer.*

- **NoSQL.** *Refers to an array of independent technologies that are meant to go beyond standard SQL to provide new access methods, generally to work with unstructured or semi-structured data.*
- **Oozie.** *A workflow engine that specializes in scheduling and managing Hadoop jobs. (Apache Software Foundation project)*
- **Open Database Connectivity (ODBC).** *A database-neutral application-programming interface (API) and related middleware that make it easy to write software that works with an expansive assortment of databases.*
- **Open Source.** *Increasingly popular, collaborative approach for developing software. As opposed to proprietary software, customers have full visibility into all source code, including the right to modify logic if necessary.*
- **Pig.** *Technology that simplifies the job of creating MapReduce applications running on Hadoop platforms. Uses a language known as 'Pig Latin'. (Apache Software Foundation project)*
- **POSIX File System.** *In the context of file systems, POSIX – which stands for Portable Operating System Interface – facilitates both random and sequential access to data. Most modern file systems are POSIX-compliant; however, the Hadoop File System is not.*
- **Scribe.** *Open source scalable technology developed at Facebook, commonly used to capture log information and write it into the Hadoop File System.*
- **Semi-structured Data.** *Information that's neither as rigidly defined as structured data (such as found in relational databases), nor as freeform as unstructured data (such as what's contained in video or audio files). XML files are a great example of semi-structured data.*
- **Snapshot.** *A read-only image of a disk volume that's taken at a particular point in time. This permits accurate rollback in situations when errors may have occurred after the snapshot was created.*
- **Spark.** *General-purpose cluster computing system, intended to simplify the job of writing massively parallel processing jobs in higher-level languages such as Java, Scala, and Python. Also includes Shark, which is Apache Hive running on the Spark platform. (Apache Software Foundation project)*
- **Structured Query Language (SQL).** *Highly popular interactive query and data manipulation language, used extensively to work with information stored in relational database management systems (RDBMS).*
- **Sqoop.** *Tool meant to ease the job of moving data - in bulk – to and from Hadoop as well as structured information repositories such as relational databases. (Apache Software Foundation project)*

- **Structured Data.** *Information that can be expressed in predictable, well-defined formats – often in the rows and columns used by relational database management systems.*
- **Tez.** *Applies and reshapes the techniques behind MapReduce to go beyond batch processing and make real-time, interactive queries achievable on mammoth data volumes. (Apache Software Foundation project)*
- **Unstructured Data.** *Information that can't be easily described or categorized using rigid, pre-defined structures. An increasingly common way of representing data, with widely divergent examples including XML, images, audio, movie clips, and so on.*
- **YARN.** *New streamlined techniques for organizing and scheduling MapReduce jobs in a Hadoop environment. (Apache Software Foundation project)*

About The Author

Robert D. Schneider is a Silicon Valley–based technology consultant and author. He has provided database optimization, distributed computing, and other technical expertise to a wide variety of enterprises in the financial, technology, and government sectors.

He has written eight books - including *Hadoop For Dummies*, published by IBM, the *Hadoop Buyers Guide*, and numerous articles on database technology and other complex topics such as cloud computing, Big Data, data analytics, and Service Oriented Architecture (SOA). He is a frequent organizer and presenter at technology industry events, worldwide. Robert blogs at www.rdschneider.com.