An incubated co-working space for technology innovation

Internal Document

**Binary Report Document**



**Table of Contents**

[An incubated co-working space for technology innovation 1](#_Toc419677051)

[Internal Document 1](#_Toc419677052)

[Document Change Log 3](#_Toc419677053)

[Definition of Terms 4](#_Toc419677054)

[Description of Acronyms iv](#_Toc419677055)

[1 Document Overview 5](#_Toc419677056)

[1.1 Introduction 5](#_Toc419677057)

[1.2 Problem Statement 6](#_Toc419677058)

[2 Proposed Solutions 7](#_Toc419677059)

[3 Architecture 8](#_Toc419677065)

[4 Tools 9](#_Toc419677065)

Document Change Log

This is a working document, which will be maintained with time. Team members, please ensure that any changes are recorded in the change log below – this is to ensure that each team member is always clear about which changes have been made and when.

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 0.01 | 24 March 2015 | Malusi Gcakasi | Created initial template document. |
| 0.02 | 24 May 2015 | Binary Group | Created reporting document. |

Definition of Terms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Central Team | Team physically located at The Cortex Hub, East London |
| Remote Team | Team physically located outside of The Cortex Hub, East London |

Chapter 1

**Introductory Phase**

Cloudera Inc. is an American-based software company that provides Apache Hadoop based software, support and services, and training to business customers. Cloudera Hadoop specialize in developing software's that allow organizations to manipulate their data easily.

The history of Big Data as a term may be brief – but many of the foundations it is built on were laid long ago. Long before computers (as we know them today) were commonplace, the idea that we were creating an ever-expanding body of knowledge ripe for analysis was popular in academia.

Although it might be easy to forget, our increasing ability to store and analyse information has been a gradual evolution – although things certainly sped up at the end of the last century, with the invention of digital storage and the Internet.

With Big Data poised to go mainstream this year, here’s a brief(ish) look at the long history of thought and innovation which have led us to the dawn of the data age.

**What is big data?**

****

Big Data means is a collection of data sets, perpetually evolving large amounts of structured, semi-structured and unstructured data and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications. It is the result of practically everything in the world being monitored and measured, creating data faster than the available technologies can store, process or manage it.

**Ancient History of Data**

C 18,000 BCE

The earliest examples we have of humans storing and analysing data are the tally sticks. The Ishango Bone was discovered in 1960 in what is now Uganda and is thought to be one of the earliest pieces of evidence of prehistoric data storage. Palaeolithic tribes people would mark notches into sticks or bones, to keep track of trading activity or supplies. They would compare sticks and notches to carry out rudimentary calculations, enabling them to make predictions such as how long their food supplies would last.

C 2400 BCE

The abacus – the first dedicated device constructed specifically for performing calculations – comes into use in Babylon. The first libraries also appeared around this time, representing our first attempts at mass data storage.

300 BC - 48 AD

The Library of Alexandria is perhaps the largest collection of data in the ancient world, housing up to perhaps half a million scrolls and covering everything we had learned so far, about pretty much everything. Unfortunately, in 48AD it is thought to have been destroyed by the invading Romans, perhaps accidentally. Contrary to common myth, not everything was lost – significant parts of the library’s collections were moved to other buildings in the city, or stolen and dispersed throughout the ancient world.

C 100 – 200 AD

The Antikythera Mechanism, the earliest discovered mechanical computer, is produced, presumably by Greek scientists. It’s “CPU” consists of 30 interlocking bronze gears and it is thought to have been designed for astrological purposes and tracking the cycle of Olympic Games. Its design suggests it is probably an evolution of an earlier device – but these so far remain undiscovered.

**The Emergence of Statistics**

1663

In London, John Graunt carries out the first recorded experiment in statistical data analysis. By recording information about mortality, he theorized that he can design an early warning system for the bubonic plague ravaging Europe.

1865

The term “business intelligence” is used by Richard Millar Devens in his Encyclopaedia of Commercial and Business Anecdotes, describing how the banker Henry Furnese achieved an advantage over competitors by collecting and analysing information relevant to his business activities in a structured manner. This is thought to be the first study of a business putting data analysis to use for commercial purposes.

1880

The US Census Bureau has a problem – it estimates that it will take it 8 years to crunch all the data collected in the 1880 census, and it is predicted that the data generated by the 1890 census will take over 10 years, meaning it will not even be ready to look at until it is outdated by the 1900 census. In 1881 a young engineer employed by the bureau – Herman Hollerith – produces what will become known as the Hollerith Tabulating Machine. Using punch cards, he reduces 10 years’ work to three months and achieves his place in history as the father of modern automated computation. The company he founds will go on to become known as IBM.

**The Early Days of Modern Data Storage**

1926

Interviewed by Colliers magazine, inventor Nikola Tesla states that when wireless technology is “perfectly applied the whole Earth will be converted into a huge brain, which in fact it is, all things being particles of a real and rhythmic whole … and the instruments through which we shall be able to do this will be amazingly simple compared to our present telephone. A man will be able to carry one in his vest pocket.”

1928

Fritz Pfleumer, a German-Austrian engineer, invents a method of storing information magnetically on tape. The principles he develops are still in use today, with the vast majority of digital data being stored magnetically on computer hard disks.

1944

Fremont Rider, librarian at Wesleyan University, Connecticut, US, published a paper titled The Scholar and the Future of the Research Library.

In one of the earliest attempts to quantify the amount of information being produced, he observes that in order to store all the academic and popular works of value being produced, American libraries would have to double their capacity every 16 years. This led him to speculate that the Yale Library, by 2040, will contain 200 million books spread over 6,000 miles of shelves.

**The Beginnings of Business Intelligence**

1958

IBM researcher Hans Peter Luhn defines Business Intelligence as “the ability to apprehend the interrelationships of presented facts in such a way as to guide action towards a desired goal.”

1962

The first steps are taken towards speech recognition, when IBM engineer William C Dersch presents the Shoebox Machine at the 1962 World Fair. It can interpret numbers and sixteen words spoken in the English language into digital information.

1964

An article in the New Statesman refers to the difficulty in managing the increasing amount of information becoming available.

**The Start of Large Data Centers**

1965

The US Government plans the world’s first data center to store 742 million tax returns and 175 million sets of fingerprints on magnetic tape.

1970

IBM mathematician Edgar F Codd presents his framework for a “relational database”. The model provides the framework that many modern data services use today, to store information in a hierarchical format, which can be accessed by anyone who knows what they are looking for. Prior to this accessing data from a computer’s memory banks usually required an expert.

1976

Material Requirements Planning (MRP) systems are becoming more commonly used across the business world, representing one of the first mainstream commercial uses of computers to speed up every day processes and make efficiencies. Until now, most people have probably only seen them in research and development or academic settings.

1989

Possibly the first use of the term Big Data (without capitalization) in the way it is used today. International best-selling author Erik Larson pens an article for Harper’s Magazine speculating on the origin of the junk mail he receives. He writes: “The keepers of big data say they are doing it for the consumer’s benefit. But data have a way of being used for purposes other originally intended.”

Additionally “business intelligence” – already a popular concept since the late 50s – sees a surge in popularity with newly emerging software and systems for analysing commercial and operational performance.

**The Internet Kicks into Gear**

1991

Computer scientist Tim Berners-Lee announced the birth of what would become the World Wide Web as we know it today. In a post in the Usenet group hypertext he sets out the specifications for a worldwide, interconnected web of data, accessible to anyone from anywhere.

1996

According to R J T Morris and B J Truskowski in their 2003 book The Evolution of Storage Systems, this is the point where digital storage became more cost effective than paper.

1997

Michael Lesk publishes his paper How Much Information is there in the World? Theorizing that the existence of 12,000 petabytes is “perhaps not an unreasonable guess”. He also points out that even at this early point in its development, the web is increasing in size 10-fold each year. Much of this data, he points out, will never be seen by anyone and therefore yield no insight.

Google Search also debuts this year – and for the next 20 years (at least) its name will become shorthand for searching the Internet for data.

**Early Ideas of Big Data**

1999

A couple of years later and the term Big Data appears in Visually Exploring Gigabyte Datasets in Real Time, published by the Association for Computing Machinery. Again the propensity for storing large amounts of data with no way of adequately analysing it is lamented. The paper goes on to quote computing pioneer Richard W Hamming as saying: “The purpose of computing is insight, not numbers.”

Also possibly first use of the term “Internet of Things”, to describe the growing number of devices online and the potential for them to communicate with each other, often without a human “middle man”. The term is used as the title of a presentation given to Procter and Gamble by RFID pioneer Kevin Ashton.

2000

In How Much Information? Peter Lyman and Hal Varian (now chief economist at Google) attempted to quantify the amount of digital information in the world, and its rate of growth, for the first time. They concluded: “The world’s total yearly production of print, film, optical and magnetic content would require roughly 1.5 billion gigabytes of storage. This is the equivalent of 250 megabytes per person for each man, woman and child on Earth.”

2001

In his paper 3D Data Management: Controlling Data Volume, Velocity and Variety Doug Laney, analyst at Gartner, defines three of what will come to be the commonly-accepted characteristics of Big Data.

This year also see the first use of the term “software as a service” – a concept fundamental to many of the cloud-based applications which are industry-standard today – in the article Strategic Backgrounder: Software as a Service by the Software and Information Industry Association.

**Web 2.0 Increases Data Volumes**

2005

Commentators announce that we are witnessing the birth of “Web 2.0” – the user-generated web where the majority of content will be provided by users of services, rather than the service providers themselves. This is achieved through integration of traditional HTML-style web pages with vast back-end databases built on SQL. 5.5 million People are already using Facebook, launched a year earlier, to upload and share their own data with friends.

This year also sees the creation of Hadoop – the open source framework created specifically for storage and analysis of Big Data sets. Its flexibility makes it particularly useful for managing the unstructured data (voice, video, raw text) which we are increasingly generating and collecting.

**Today’s Use of the Term ‘Big Data’ Emerges**

2007

Wired brings the concept of Big Data to the masses with their article The End of Theory: The Data Deluge Makes the Scientific Model Obsolete.

2008

The world’s servers process 9.57 zettabytes (9.57 trillion gigabytes) of information – equivalent to 12 gigabytes of information per person, per day), according to the How Much Information? 2010 report. In International Production and Dissemination of Information, it is estimated that 14.7 exabytes of new information are produced this year.

2009

The average US Company with over 1,000 employees is storing more than 200 terabytes of data according to the report Big Data: The Next Frontier for Innovation, Competition and Productivity by McKinsey Global Institute.

2010

Eric Schmidt, executive chairman of Google, tells a conference that as much data is now being created every two days, as was created from the beginning of human civilization to the year 2003.

2011

The McKinsey report states that by 2018 the US will face a shortfall of between 140,000 and 190,000 professional data scientists, and states that issues including privacy, security and intellectual property will have to be resolved before the full value of Big Data will be realized.

2014

The rise of the mobile machines – as for the first time, more people are using mobile devices to access digital data, than office or home computers. 88% of business executives surveyed by GE working with Accenture report that big data analytics is a top priority for their business.

Final Thought

What this teaches us is that Big Data is not a new or isolated phenomenon, but one that is part of a long evolution of capturing and using data. Like other key developments in data storage, data processing and the Internet, Big Data is just a further step that will bring change to the way we run business and society. At the same time it will lay the foundations on which many evolutions will be built.

As always, I am keen to hear your thoughts on the topic, please share them in the comments below. For example, are there other mile-stones you would have included?

**1.2 Challenges**

<https://www.progress.com/~/media/Progress/Documents/Papers/Addressing-Five-Emerging-Challenges-of-Big-Data.pdf>

**1.3 Examples of what you can accomplish with big data**

Dialogue with consumers

Today’s consumers are not easily pleased. They look around a lot before they buy, talk to their entire social network about their purchases, demand to be treated as unique and want to be sincerely thanked for buying your products. Big Data allows you to profile these increasingly vocal and fickle little ‘tyrants’ in a far-reaching manner so that you can engage in an almost one-on-one, real-time conversation with them. This is not actually a luxury. If you don’t treat them like they want to, they will leave you in the blink of an eye.

Just a small example: when any customer enters a bank, Big Data tools allow the clerk to check his/her profile in real-time and learn which relevant products or services (s) he might advise. Big Data will also have a key role to play in uniting the digital and physical shopping spheres: a retailer could suggest an offer on a mobile carrier, on the basis of a consumer indicating a certain need in the social media.

Re-develop your products

Big Data can also help you understand how others perceive your products so that you can adapt them, or your marketing, if need be. Analysis of unstructured social media text allows you to uncover the sentiments of your customers and even segment those in different geographical locations or among different demographic groups.

On top of that, Big Data lets you test thousands of different variations of computer-aided designs in the blink of an eye so that you can check how minor changes in, for instance, material affect costs, lead times and performance. You can then raise the efficiency of the production process accordingly.

Perform risk analysis

Success not only depends on how you run your company. Social and economic factors are crucial for your accomplishments as well. Predictive analytics, fuelled by Big Data allows you to scan and analyse newspaper reports or social media feeds so that you permanently keep up to speed on the latest developments in your industry and its environment. Detailed health-tests on your suppliers and customers are another goodie that comes with Big Data. This will allow you to take action when one of them is in risk of defaulting.

Keeping your data safe

You can map the entire data landscape across your company with Big Data tools, thus allowing you to analyse the threats that you face internally. You will be able to detect potentially sensitive information that is not protected in an appropriate manner and make sure it is stored according to regulatory requirements. With real-time Big Data analytics you can, for example, flag up any situation where 16 digit numbers – potentially credit card data - are stored or emailed out and investigate accordingly.

Create new revenue streams

The insights that you gain from analysing your market and its consumers with Big Data are not just valuable to you. You could sell them as non-personalized trend data to large industry players operating in the same segment as you and create a whole new revenue stream.

One of the more impressive examples comes from Shazam, the song identification application. It helps record labels find out where music sub-cultures are arising by monitoring the use of its service, including the location data that mobile devices so conveniently provide. The record labels can then find and sign up promising new artists or remarket their existing ones accordingly.

Customize your website in real time

Big Data analytics allows you to personalize the content or look and feel of your website in real time to suit each consumer entering your website, depending on, for instance, their sex, nationality or from where they ended up on your site. The best-known example is probably offering tailored recommendations: Amazon’s use of real-time, item-based, collaborative filtering (IBCF) to fuel its ‛Frequently bought together’ and ‛Customers who bought this item also bought’ features or Linked In suggesting ‛People you may know’ or ‛Companies you may want to follow’. And the approach works: Amazon generates about 20% more revenue via this method.

Reducing maintenance costs

Traditionally, factories estimate that a certain type of equipment is likely to wear out after so many years. Consequently, they replace every piece of that technology within that many years, even devices that have much more useful life left in them. Big Data tools do away with such unpractical and costly averages. The massive amounts of data that they access and use and their unequalled speed can spot failing grid devices and predict when they will give out. The result: a much more cost-effective replacement strategy for the utility and less downtime, as faulty devices are tracked a lot faster.

Offering tailored health-care

We are living in a hyper-personalized world, but health-care seems to be one of the last sectors still using generalized approaches. When someone is diagnosed with cancer they usually undergo one therapy, and if that doesn’t work, the doctors try another, etc. But what if a cancer patient could receive medication that is tailored to his individual genes? This would result in a better outcome, less cost, less frustration and less fear.

With human genome mapping and Big Data tools, it will soon be commonplace for everyone to have their genes mapped as part of their medical record. This brings medicine closer than ever to finding the genetic determinants that cause a disease and developing drugs expressly tailored to treat those causes — in other words, personalized medicine.

Offering enterprise-wide insights

Previously, if business users needed to analyse large amounts of varied data, they had to ask their IT colleagues for help as they themselves lacked the technical skills for doing so. Often, by the time they received the requested information, it was no longer useful or even correct. With Big Data tools, the technical teams can do the groundwork and then build repeatability into algorithms for faster searches. In other words, they can develop systems and install interactive and dynamic visualization tools that allow business users to analyse, view and benefit from the data.

<http://datascienceseries.com/stories/ten-practical-big-data-benefits>

**Chapter 2**

**Computer storage**

Computer data storage is a technology consisting of computer components used to retain digital data.

1920

The storage era started in the early 1928 where the magnetic tape was introduced. The inventor of this storage device was a German engineer, Fritz Pfleumer.

1930

G. Taushek an Austrian innovator, invented the magnetic drum in 1932. He based his invention off a discovery credited to Fritz Pfleumer.

1940

Professor Fredrick C. Williams and his colleagues developed the first random access computer memory at the University of Manchester located in the United Kingdom. He used a series of electrostatic cathode-ray tubes for digital storage. A storage of 1024 bits of information was successfully implemented in 1948.

1971

IBM started its development of an inexpensive system geared towards loading microcode into the System/370 mainframes. As a result, the 8-inch floppy emerged. A floppy disk, a portable storage device made of magnetic film encased in plastic, made it easier and faster to store data.

1995

DVD became the next generation of digital disc storage. DVD, a bigger and faster alternative to the compact disc, serves to store multimedia data.

Current

Cloud enables businesses to protect data using backup, recover from a disaster, and archive unused files using only a lightweight software client. As storage hardware and internet bandwidth continue to develop, so will the cloud computing.

Conclusion

The need for large amounts of storage devices grows daily, with individual users, organization, social networks generate huge amounts of data.

Chapter 3

**Processing**

Data processing is the Manipulation of data by a computer. The processing range from the conversion of raw data to machine-readable form, flow of data through the CPU and memory to output devices, and formatting or transformation of output.

In nowadays data processing is very critical to supporting organization's everyday operations such as generating reports for suppliers and customers and measuring internal metrics day to day.



Processing large and complex data sets that cannot be stored on a single machine becomes a challenge for traditional data processing applications. Challenges include analysis, capture, data curation [activities required to maintain research data long-term such that is available for reuse], search, sharing, storage, transfer, visualization, and information privacy.

**Problems**

Flexibility and Data Quality Constraints - Raw data is not perfect: source systems can be misconfigured, reporting formats change and third party data sources may contain mistakes. Traditional data quality processes look at individual records using fixed rules for identifying errors, and the errors are kicked out or corrected according to rules after which processing continues.

Too much time - While limiting flexibility and hindering data quality, the systems in place to extract, transform and load that data into the warehouse have become a bottleneck as data volumes have exploded. Because unstructured data must be reformatted to fit into a relational schema before it can be loaded into the system, it requires an extra data processing step that slows ingestion, creates latency and eliminates elements of the data that could become important down the road.

Too much data - The data processing infrastructure that was developed to run business operations over the past decades is having trouble keeping pace with today’s digital landscape. Across every industry, daily interactions and transactions are moving online and business services are becoming increasingly automated. Volumes of multi-structured, machine-generated data from a variety of sources have skyrocketed, and smart companies want to capture and make use of it all.

**Solutions**

Flexibility- The underlying storage in Hadoop is a flexible file system that can hold any type of data. Because of this, Hadoop can accept any kind of file in any format and store this raw data in perpetuity. Hadoop supports pluggable serialization so it can be used to efficiently and reliably store raw data in its original format.

Scalability- Hadoop leverages a scale-out architecture that marries self-healing, high-bandwidth clustered storage (via the Hadoop Distributed File System, or HDFS) with fault-tolerant distributed processing (MapReduce). HDFS is structured much like a regular network file system and is

Optimized for large scale data collection and processing. It is linearly scalable — processing power and storage capacity scale linearly as additional industry standard servers are incorporated.

HDFS and MapReduce are both designed to run on scale out systems. The data loaded into HDFS is partitioned sequentially across any number of standard servers with local disk. This makes provisioning new servers very efficient and requires no additional effort on the part of the developer to handle massive changes in data volume.

Data Integrity- The Hadoop processing framework, MapReduce, offers a robust application program interface (API) that allows both record-at-a-time processing for basic quality checks and single-pass, multi-group processing for efficient execution of more robust quality checks across records. MapReduce includes built-in mechanisms for handling data errors and for detecting large numbers of errors (such as when a data set has changed). This early detection gives operators a chance to identify and quickly resolve data processing challenges.

Affordability- Because Hadoop uses industry standard hardware, the cost per terabyte of storage is, on average, 10x cheaper than a traditional relational data warehouse system. Hadoop uses standard servers with local storage, thereby optimizing for high I/O workloads. Servers are connected using standard gigabit and 10 gigabit networking, which lowers overall system cost and still allows near limitless storage and processing by scaling out.

**Chapter 4**

Hadoop Cloudera

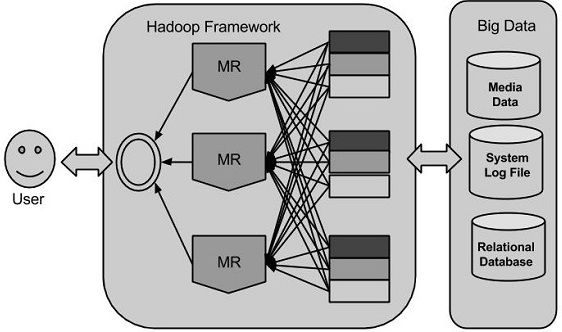


**Doug Cutting**, Cloudera's Chief Architect, helped create Apache Hadoop out of necessity as data from the web exploded, and grew far beyond the ability of traditional systems to handle it. Hadoop was initially inspired by papers published by Google outlining its approach to handling an avalanche of data, and has since become the standard for storing, processing and analysing hundreds of terabytes, and even petabytes of data.



Apache Hadoop is 100% open source, and pioneered a fundamentally new way of storing and processing data. Instead of relying on expensive, proprietary hardware and different systems to store and process data, Hadoop enables distributed parallel processing of huge amounts of data across inexpensive, industry-standard servers that both store and process the data, and can scale without limits. With Hadoop, no data is too big. And in today’s hyper-connected world where more and more data is being created every day, Hadoop’s breakthrough advantages mean that businesses and organizations can now find value in data that was recently considered useless.

**Hadoop Framework**



Origins of Hadoop?

The underlying technology was invented by Google back in their earlier days so they could usefully index all the rich textural and structural information they were collecting, and then present meaningful and actionable results to users. There was nothing on the market that would let them do that, so they built their own platform. Google’s innovations were incorporated into “Nutch”, an open source project, and Hadoop was later spun-off from that. Yahoo has played a key role developing Hadoop for enterprise applications.

What problems can Hadoop solve?

The Hadoop platform was designed to solve problems where you have a lot of data — perhaps a mixture of complex and structured data — and it doesn’t fit nicely into tables. It’s for situations where you want to run analytics that are deep and computationally extensive, like clustering and targeting. That’s exactly what Google was doing when it was indexing the web and examining user behaviour to improve performance algorithms.

Hadoop applies to a bunch of markets. In finance, if you want to do accurate portfolio evaluation and risk analysis, you can build sophisticated models that are hard to jam into a database engine. But Hadoop can handle it. In online retail, if you want to deliver better search answers to your customers so they’re more likely to buy the thing you show them, that sort of problem is well addressed by the platform Google built. Those are just a few examples.

How is Hadoop architected?

Hadoop is designed to run on a large number of machines that don’t share any memory or disks. That means you can buy a whole bunch of commodity servers, slap them in a rack, and run the Hadoop software on each one. When you want to load all of your organization’s data into Hadoop, what the software does is bust that data into pieces that it then spreads across your different servers. There’s no one place where you go to talk to all of your data; Hadoop keeps track of where the data resides. And because there are multiple copy stores, data stored on a server that goes off line or dies can be automatically replicated from a known good copy.

In a centralized database system, you’ve got one big disk connected to four or eight or 16 big processors. But that is as much horsepower as you can bring to bear. In a Hadoop cluster, every one of those servers has two or four or eight CPUs. You can run your indexing job by sending your code to each of the dozens of servers in your cluster, and each server operates on its own little piece of the data. Results are then delivered back to you in a unified whole. That’s "MapReduce”: you map the operation out to all of those servers and then you reduce the results back into a single result set.

Architecturally, the reason you’re able to deal with lots of data is because Hadoop spreads it out. And the reason you’re able to ask complicated computational questions is because you’ve got all of these processors, working in parallel, harnessed together.

Core Hadoop Two Main Systems

There are two key functional components within this ecosystem: The storage of data (Hadoop Distributed File System File System, or HDFS) and the framework for running parallel computations on this data (MapReduce).

Hadoop Distributed File System (HDSF) self-healing high-bandwidth clustered storage. MapReduce distributes fault-tolerant resource management and scheduling coupled with a scalable data programming abstraction.

**HDFS Hadoop Distributed File System**

HDFS is the “secret sauce” that enables Hadoop to store huge files. It’s a scalable file system that distributes and stores data across all machines in a Hadoop cluster (a group of servers). Each HDFS cluster contains the following:

NameNode: Runs on a “master node” that tracks and directs the storage of the cluster.

DataNode: Runs on “slave nodes,” which make up the majority of the machines within a cluster. The NameNode instructs data files to be split into blocks, each of which are replicated three times and stored on machines across the cluster. These replicas ensure the entire system won’t go down if one server fails or is taken off line—known as “fault tolerance.”

Client machine: neither a NameNode nor a DataNode, Client machines have Hadoop installed on them. They’re responsible for loading data into the cluster, submitting MapReduce jobs and viewing the results of the job once complete.

**MapReduce Computational Framework**

MapReduce is the system used to efficiently process the large amount of data Hadoop stores in HDFS. Originally created by Google, its strength lies in the ability to divide a single large data processing job into smaller tasks. All MapReduce jobs are written in Java, but other languages can be used via the Hadoop Streaming API, which is a utility that comes with Hadoop.

Once the tasks have been created, they’re spread across multiple nodes and run simultaneously (the “map” step). The “reduce” phase combines the results together. This delegation of tasks is handled by two “daemons,” the JobTracker and TaskTracker. The technical definition of a daemon is “a process that is long-lived.” In our house example, a daemon can be thought of as a foreman: the jobs may change (new houses must be built), workers will come and go, but the foreman is always there to oversee the job and delegate tasks.

JobTracker: The JobTracker oversees how MapReduce jobs are split up into tasks and divided among nodes within the cluster.

TaskTracker: The TaskTracker accepts tasks from the JobTracker, performs the work and alerts the JobTracker once it’s done. TaskTrackers and DataNodes are located on the same nodes to improve performance.

MapReduce Resource

Manager/ Scheduler

A given job is broken down into tasks, then tasks are scheduled to be as close to data as possible. Three levels of data locality same server as

A given job is broken down into tasks, then tasks are scheduled to be as close to data as possible.

Three levels of data locality same server as data (local disk), same rack as data (rack/leaf switch), wherever there is a free slot (cross rack)

Optimized for batch processing, failure recovery.

System detects laggard tasks and speculatively executes parallel tasks on the same slice of data.

Data locality: An important concept with HDFS and MapReduce, data locality can best be described as “bringing the compute to the data.” In other words, whenever you use MapReduce program on a particular part of HDFS data, you always want to run that program on the node, or machine, that actually stores this data in HDFS. Doing so allows processes to be run much faster, since it prevents you from having to move large amounts of data around.

When a MapReduce job is submitted, part of what the JobTracker does is look to see which machines the blocks required for the task are located on. This is why, when the NameNode splits data files into blocks, each one is replicated three times: the first is stored on the same machine as the block, while the second and third are each stored on separate machines.

Storing the data across three machines thus gives you a much higher chance of achieving data locality, since it’s likely that at least one of the machines will be freed up enough to process the data stored at that particular location.

Yet another Resource Negotiator (YARN)

YARN is an updated way of handling the delegation of resources for MapReduce jobs. It takes the place of the JobTracker and TaskTracker. In our house example, if JobTracker and TaskTracker can be thought of as the foreman, YARN is a foreman with an MBA—it’s a more advanced way of carrying out MapReduce jobs.

It also gives you added abilities, such as the ability to work with frameworks other than MapReduce and to translate jobs developed in languages other than Java.



HBase

HBase is a columnar database management system that is built on top of Hadoop and runs on HDFS. Like MapReduce, HBase applications are written in Java, as well as other languages via their Thrift database, which is a framework that allows cross-language services development. The key difference between MapReduce and HBase is that HBase is intended to work with random workloads.

For example, if you have regular files that need to be processed, MapReduce works just fine. But if you have a table that is a petabyte in size and you need to process a single row from a random location within this table, you would use HBase. Another benefit of HBase is the extremely low latency, or time delay, it provides.



It’s important to note, however, that HBase and MapReduce are not mutually exclusive. In fact, you can often run them together—MapReduce can run against an HBase table or a file, for example.

Hive

MapReduce jobs are often written in Java. But not everyone using Hadoop knows Java—the preferred syntax is SQL, which is essentially the “lingua franca” between all programming languages in the BI/big data space.

Hive allows users who aren’t familiar with programming to access and analyse big data in a less technical way, using a SQL-like syntax called Hive Query Language (HiveQL). HiveQL is used to create programs that run just like MapReduce would on a cluster.

In a very general sense, Hive is used for complex, long-running tasks and analyses on large sets of data, e.g. analysing the performance of every store within a particular region for a chain retailer.



Impala

Like Hive, Impala also uses SQL syntax instead of Java to access data. What distinguishes Hive and Impala is speed: While a query using Hive may take minutes, hours or longer, a query using Impala usually take seconds (or less).

Impala is used for analyses that you want to run and return quickly on a small subset of your data, e.g. analysing company finances for a daily or weekly report. Since Impala is meant to be used as an analytic tool on top of prepared, more structured data, it’s not ideal if you’re in the process of data preparation and complex data manipulation, e.g. ingesting raw machine data from log files.

A good way to think about Hive and Impala is to compare them to a screwdriver and drill bit: both can do the same—or similar—jobs, but the drill (Impala) is much faster.



Apache Pig

Like Hive and Impala, Pig is a high-level platform used for creating MapReduce programs more easily. The programming language Pig uses is called Pig Latin, and it allows you to extract, transform and load (ETL) data at a very high level—meaning something that would require several hundred lines of Java code can be expressed in, say, 10 lines of Pig.

While Hive and Impala require data to be more structured in order to be analysed, Pig allows you to work with unstructured data. In other words, while Hive and Impala are essentially query engines used for more straightforward analysis, Pig’s ETL capability means it can perform “grunt work” on unstructured data, cleaning it up and organizing it so that queries can be run against it.



Hadoop Common

Usually only referred to by programmers, Hadoop Common is a common utilities library that contains code to support some of the other modules within the Hadoop ecosystem. When Hive and HBase want to access HDFS, for example, they do so using JARs (Java archives), which are libraries of Java code stored in Hadoop Common.

Apache Spark

While not yet part of the Hadoop ecosystem, Apache Spark is frequently mentioned along with Hadoop, so we’ll take a moment to touch on it here. Spark is an alternative way to perform the type of batch-oriented processing that MapReduce does. (Batch-oriented means that it will take a certain amount of time for a result to be returned, as opposed to returning it in real-time.)

While MapReduce jobs use data that have been replicated and stored on-disk within a cluster, Spark allows you to leverage the memory space on servers, performing in-memory computing. This allows for real-time data processing that is up to 100 times faster than MapReduce in some instances.

**Chapter 5**

**Solution Engineering**

**Step 1: Understand how the organization makes money**

Need to pretend as the general manager of the organization. Look at factors such as what an organization do to increase revenues, decrease costs, reduce risks, or increase compliance? Spend the time to identify and understand your organization’s strategic nouns, and ascertain how those nouns drive the money making capabilities of the organization. Invest time actually using your organization’s products or services. Experience first-hand how your organization’s product or products work.

**Step 2: Identity your organization’s key business initiatives**

This step focus on doing some primary research to understand your organization’s key business initiatives. This includes reading the annual report, listening to analyst calls, and researching for recent executive management speeches and presentations. Make interviews with senior business management to understand their business initiatives and opportunities, as well as their perceptions of the key challenges that might prevent the organization from successfully executing against their top business opportunities.

**Step 3: Brainstorm Big Data Business impact**

In this step the developers must brainstorm how big data and advanced analytics can impact the targeted business initiative.

Mine the more detailed transactional data at the lowest level of transaction granularity, which enables more granular and detailed decisions. For example, analyse the detailed transactional data, such as customer loyalty transactions, to enable more granular decision making and uncover new data monetization opportunities at the individual customer, seasonal/holiday, and location/geography levels.

Integrate new unstructured data sources to enable more robust and complete decisions. This includes internal unstructured data sources such as consumer comments, call center notes, e-mail, physician notes, and services bay logs, as well as external unstructured data sources such as social media posts, blogs, mobile/smartphone apps and third-party or public data sources.

Provide real-time/low-latency data access where you reduce the time delay between when the data event occurs and the analysis of that data event, which enables more frequent and timely decisions and data monetization.

Integrate predictive analytics into your key business process to provide new opportunities to uncover causality buried in the data. Predictive analysis enable a different mindset with your business stakeholders, encouraging them to use new verbs.

**Step 4: Break down the business initiative into use cases**

Consists of conducting a series of interviews and ideation/envisioning workshops to brainstorm, identify, define, aggregate, and prioritize the use cases necessary to support the targeted business initiative.

Targeted personas and stakeholders, including their roles, responsibilities, and expectations. Business questions the business stakeholders are trying to answer, or could be trying to answer if they had access to more detailed and diverse data sources.

Business decisions the business stakeholders are trying to make, and the supporting decision processes including timing, decision flow/process, and downstream stakeholders.

**Step 5: Prove out the use case**

Deploy data and technology to validate the analytic feasibility of the solution. Introduce a Proof of Value analytic lab to prove out the business case (financial model, ROI and analytic lift) using the full depth of available data and full depth of available data and full breath of available technology capabilities.

**Step 6: Design and implement the big data solution**

Based on the Proof of Value analytic lab process, it’s now time to start defining and building the detailed data models, analytic models, technology architectures, and production roadmap for integrating the analytic models and insights.

Data sources and data access requirements: is a detailed plan and roadmap for prioritizing what data to capture and where to store that. The plan should address both structured and unstructured data.

Instrumentation strategy: It is likely that additional data about your customers, products, and operations will need to captured, mainly out of the existing business processes.

Real-time data access and analysis requirements: Certain use cases are going to require real-time (or low-latency) data access, analysis, and decision making as data is flowing through the business. These real-time requirements must be addressed across in the entire technology and architectural stack including your Extract, Transform and load and Extract, load, and Transform algorithms, data transformation and enrichment process, in-memory computing, complex event processing, data platform, analytic models, and user experience.

Data modelling capabilities: Data modelling requirements need to encompass all the traditional data warehousing architectural approaches- operational data store, data staging area, data marts, enterprise data warehouse-plus many of the new data platform and data federation tools and techniques that are available. The data modelling plan need to consider data schema design and the role of NoSQL databases, Hadoop and the Hadoop Distributed File System.

Business intelligence: Most organizations have an existing business intelligence environment in place that address key performance indicators, reporting, alerts and dashboard requirements. Now here it’s where one determine how to enhance that investment with new big data capabilities such as unstructured data, real – time data feeds, and predictive analytics.

User experience requirements: there must be a user experience plan, it needs to include the wireframe and mock-up processes to ensure an understanding of how the analytic results and models will manifest themselves into the business user’s daily operations and the management reports and dashboards.

**Chapter 6**

**Code snippets**

**MapReduce**

package org.myorg;

import java.io.IOException;

import java.util.\*;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.conf.\*;

import org.apache.hadoop.io.\*;

import org.apache.hadoop.mapred.\*;

import org.apache.hadoop.util.\*;

public class WordCount {

public static class Map extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {

private final static IntWritable one = new IntWritable(1);

private Text word = new Text();

public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {

String line = value.toString();

StringTokenizer tokenizer = new StringTokenizer(line);

while (tokenizer.hasMoreTokens()) {

word.set(tokenizer.nextToken());

output.collect(word, one);

}

}

}

public static class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> { public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException { int sum = 0;

while (values.hasNext()) {

sum += values.next().get();

}

output.collect(key, new IntWritable(sum));

}

}

public static void main(String[] args) throws Exception {

JobConf conf = new JobConf(WordCount.class);

conf.setJobName("wordcount");

conf.setOutputKeyClass(Text.class);

conf.setOutputValueClass(IntWritable.class);

conf.setMapperClass(Map.class);

conf.setCombinerClass(Reduce.class);

conf.setReducerClass(Reduce.class);

conf.setInputFormat(TextInputFormat.class);

conf.setOutputFormat(TextOutputFormat.class);

FileInputFormat.setInputPaths(conf, new Path(args[0]));

FileOutputFormat.setOutputPath(conf, new Path(args[1]));

JobClient.runJob(conf);

}

}

**Apache Mahout**

package org.acme;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.FileReader;

import java.util.List;

import org.apache.hadoop.fs.Path;

import org.apache.mahout.classifier.ClassifierResult;

import org.apache.mahout.classifier.bayes.TrainClassifier;

import org.apache.mahout.classifier.bayes.algorithm.BayesAlgorithm;

import org.apache.mahout.classifier.bayes.common.BayesParameters;

import org.apache.mahout.classifier.bayes.datastore.InMemoryBayesDatastore;

import org.apache.mahout.classifier.bayes.exceptions.InvalidDatastoreException;

import org.apache.mahout.classifier.bayes.interfaces.Algorithm;

import org.apache.mahout.classifier.bayes.interfaces.Datastore;

import org.apache.mahout.classifier.bayes.model.ClassifierContext;

import org.apache.mahout.common.nlp.NGrams;

public class Starter {

public static void main( final String[] args ) {

final BayesParameters params = new BayesParameters();

params.setGramSize( 1 );

params.set( "verbose", "true" );

params.set( "classifierType", "bayes" );

params.set( "defaultCat", "OTHER" );

params.set( "encoding", "UTF-8" );

params.set( "alpha\_i", "1.0" );

params.set( "dataSource", "hdfs" );

params.set( "basePath", "/tmp/output" );

try {

Path input = new Path( "/tmp/input" );

TrainClassifier.trainNaiveBayes( input, "/tmp/output", params );

Algorithm algorithm = new BayesAlgorithm();

Datastore datastore = new InMemoryBayesDatastore( params );

ClassifierContext classifier = new ClassifierContext( algorithm, datastore );

classifier.initialize();

final BufferedReader reader = new BufferedReader( new FileReader( args[ 0 ] ) );

String entry = reader.readLine();

while( entry != null ) {

List< String > document = new NGrams( entry,

Integer.parseInt( params.get( "gramSize" ) ) )

.generateNGramsWithoutLabel();

ClassifierResult result = classifier.classifyDocument(

document.toArray( new String[ document.size() ] ),

params.get( "defaultCat" ) );

entry = reader.readLine();

}

} catch( final IOException ex ) {

ex.printStackTrace();

} catch( final InvalidDatastoreException ex ) {

ex.printStackTrace();

}

}

**Referencing**

Bill, S. (2013), *Big Data: Understanding How Data Powers Big Business.* Canada: John Wiley & Sons, Inc.