Unit 12 Notes

Table of Contents

Introduction to Big Data ? 1
And The Most Valuable Asset of Any Organization Is ? 2
The Information Needs of the Organization 2
A Data Explosion 3
Business Move to the Web 3
Social Revolution 3
Mobile Madness 4
Internet of Things 4
Too Much Information 5
What Makes Data "Big" ? 5
Other factors 6
What Types of Data are Big Data? 7
Summary 7
References .... **Error! Bookmark not defined.**
Introduction to Hadoop 7
What is Hadoop ? 7
Why is Something Like This Needed? 8
How it works 9
Architecture 9
HDFS 10
Map-Reduce and YARN 11
Components of the Hadoop Ecosystem 12
Essential Tools 12
Other Tools 13
References .... **Error! Bookmark not defined.**

Introduction to Big Data

What exactly do we mean by "Big Data"? What makes big data "big"? Before we get there, let's go back to the basics...

And The Most Valuable Asset of Any Organization Is ?

The most valuable asset of any organization is its data1. You might be thinking: "Isn't it the employees? Or the intellectual property? or the business plan?" Certainly you cannot run a successful organization without those things, but to excel, be better than the competition, and truly innovate you must harness the power of data. All organizations have data. All organizations use data. Its the degree that your organization:

- understands its own data and the data which influences your business,
- make use of data to influence decision making to mitigate risk and discover new insights, and
- uses data measure goals and initiatives to determine success

which ultimately determine how valuable data is to your organization2.

To clarify, let's look at a fictitious retailer, Fudgemart. Fudgemart sells a variety of consumer goods through online and brick-and-mortar stores. No product is outside the scope of Fudgemart - they sell everything from hardware to clothing to consumer electronics.

The Information Needs of the Organization

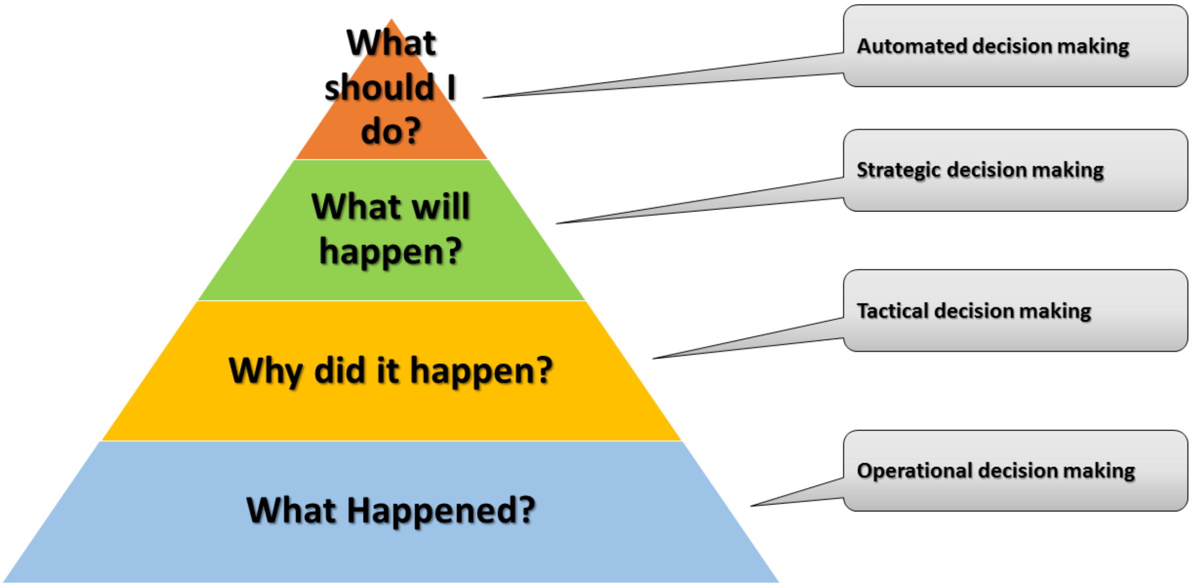
At the most basic level, Fudgemart uses its data to answer the question "What Happened?" There are sales reports detailing items sold and returned. Payroll data lists the amount of money spent on salaries. Month end reports project sales totals to goals. Quarterly reports outline the fiscal health of the company to its shareholders. These reports address the most basic informational need of the organization.

Given these reports, it is natural for decision-makers to start asking more demanding questions of their data, like "Which customers are likely to purchase snow shovels this winter?" or "What will our sales be like in 6 months?" Fudgemart uses data mining and machine learning algorithms to forecast sales, streamline inventory levels and better understand the purchasing habits of its customers. Fudgemart's data is used to identify trends and patterns; to influence intelligent decision making.

When Fudgemart really starts to understand its data, they will start automating decisions and taking actions based on data. For example, Fudgemart knows northeastern customers purchase snow shovels in the winter months thus, during these months they automatically re-supply these stores with shovels from their supplier as they are purchased by the customer. This minimizes excessive inventory, meets customer demand and therefore lowers costs.

1 https://www.linkedin.com/pulse/20141119183044-243338813-data-your-most-valuable-asset-a-business-case-for-data-governance

2 http://iveybusinessjournal.com/publication/why-big-data-is-the-new-competitive-advantage/



Fudgemart is not doing anything revolutionary. They're taking the data driving the business, known as business /corporate data, then storing, re-structuring, and analyzing it to gain informational insights. In fact organizations have built decision-aiding tools like enterprise data warehouses, analytic applications, and decision support systems from their business data for decades.

A Data Explosion

Around the turn of the millennium began a major disruption of the traditional data landscape, courtesy of the Internet. The Internet has grown seemingly exponentially from a network exclusive to academics, governments and researchers into critical infrastructure for the ubiquitous networking of everything. It ushered in a gluttony of new data sources and forced organizations us to re-imagine what is considered business data and how we go about obtaining insights from it.

Business Move to the Web

It started with websites in the 1990's. Using website log data, organizations could track movement of users as they click through their website, using the data to understand current customer behavior and target new ones. The next logical step from websites was e-commerce. Companies put business data online, accepting orders for goods and services. From this data companies can learn more about customer habits, such as products they view but do not purchase. Companies can apply machine learning to spending habits and recommend new products to customers in their web browsers. As competing companies join the e-commerce revolution and add their product catalogs online it becomes easy to scrape web sites for competitor data. Suppliers move online, too. They facilitate business to business communication over the web,offering service-oriented architecture. Companies can automate the supply chain, querying supplier inventory and placing orders in real-time.

Social Revolution

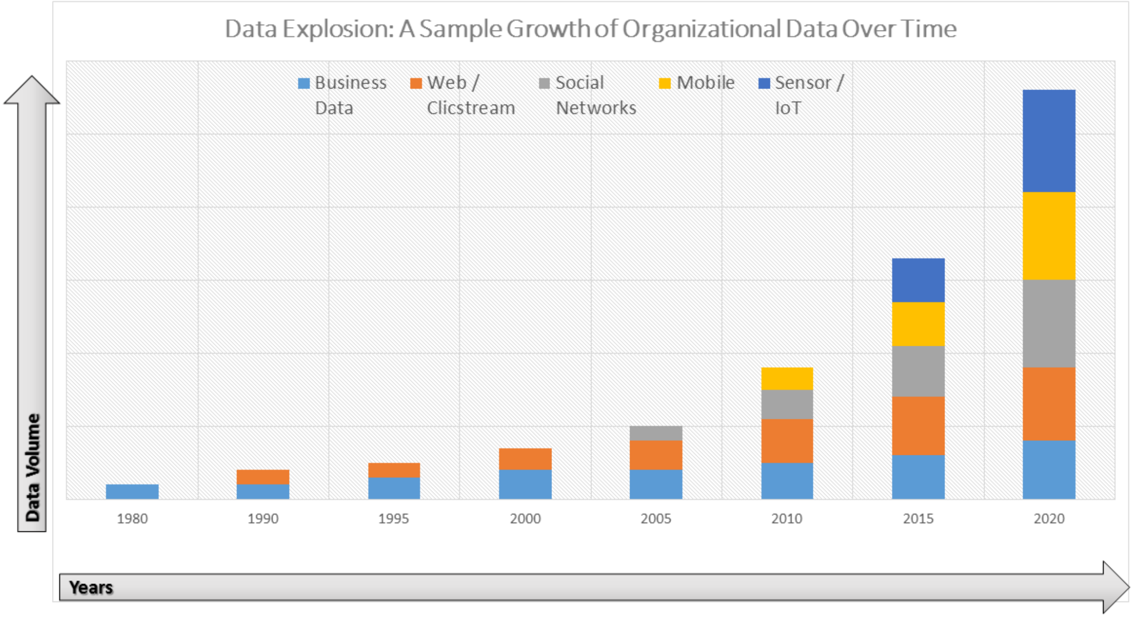
Before social networking sites, if you wanted an online presence you required knowledge of how to build and host a website. Sites like MySpace, YouTube, Facebook, Linked in and Twitter changed the game, making it easy for non-technologists to share their lives online. Organizations join the fray, engaging customers through social media as an extension of customer relations, marketing and customer service. Organizations see the potential of this medium beyond the obvious of mining it for sentiment of what others are saying about their brands and products. Companies use the data to build extensive profiles of their customer's work, hobbies, and habits with aspirations of being able to identify new customers, products or markets. For example, through social media companies might learn you are a musician and target you with offers to buy products accordingly.

Mobile Madness

Smart phones usher in a new level of convenience for users - now they can update social media and shop online from just about anywhere. When customers browse websites or use their app from a mobile phone, companies have access to geographic location and sensor data. They can use this data to determine if a customer is in one of their stores or a competitors, for example. Location data, revealing where someone is at a point in time, can be combined with weather and other data sources to determine other factors which might influence behavior. Companies reward customers for being there. The convenience of smart phones leads to increase in usage sending the data collection volumes to new levels.

Internet of Things

A world of network connected devices and sensors has far-reaching implications for organizations. Shipping companies can monitor the driving habits of truck drivers detecting for signs of driver fatigue. Power companies using smart windmills get operational status reports and feedback from wind farms. Retailers can get a sense of foot traffic patterns, how customers walk through a store, where they stop and for how long. There's huge upside to the Internet of Things but also a massive amount of data being generated from sensors that which must be collected, stored, then analyzed in real-time.



Too Much Information

Organizations want to harness this new data, integrate it with their business data and explore how it can be used to influence decision-making. However, there's a fundamental problem: since the volume, speed and complexity of the data outpaces the computing power of the traditional enterprise data warehouse, data cannot be easily processed with traditional techniques. We need a new way to store, process and analyze this "big" data.

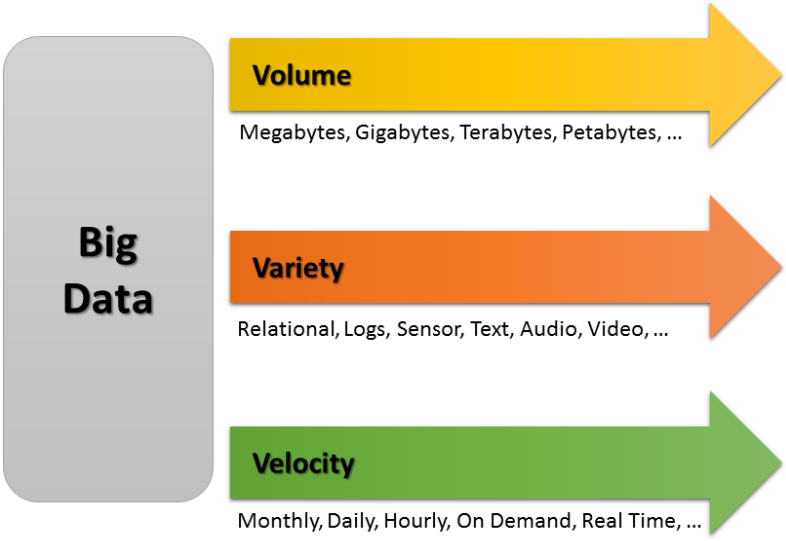
What Makes Data "Big" ?

It is important to recognize that the term "Big Data" does not simply imply that there's a large quantity of data. For years, enterprises have been dealing with large volumes of data, so while size is a factor is is not the only one.

The industry generally accepts three base factors for what makes up big data as a definition 3. Big data has all three of these characteristics, not just a single factor.

1. **volume**: As one might expect large quantities of data factor into the definition. Web site logs, social network feeds and sensor data each occur in large volumes. A key factor with big data volume is that the data are not confined to a single system, but is rather distributed across the organization, making it impractical and sometimes impossible to load all data into a single system for processing.
2. **velocity**: Velocity refers to the rate of change of the data. If we want to make automated decisions from our data it needs to be processed as close to real-time as possible. This is not a great challenge with business data living in an enterprise database with transaction management. When we're making decisions from log files, sensor data, or social media we require a computationally-intensive level of processing that often cannot be met through traditional means.
3. **variety**: Variety addresses the types of data. Business data is often well structured and stored in a RBDMS (Relational Database Management System) New data like web logs, sensor data, audio, video, and social media are semi-structured or unstructured. These types of data are not conducive to RDBMS storage, thus we need a different way to store this data so it is can be easily accessible.

[3] http://www.datasciencecentral.com/forum/topics/the-3vs-that-define-big-data


Big Data
Volume
Megabytes, Gigabytes, Terabytes, Petabytes, ...
Variety
Relational, Logs, Sensor, Text, Audio, Video, ...
Velocity
Monthly, Daily, Hourly, On Demand, Real Time, ...

Let's go back to Fudgemart for an example. Fudgemart has business data about customer online orders in its enterprise database, a RDBMS. This data is well structured into database tables, and there's a real-time dashboard of customer information as orders take place, including a map of where the order will be shipped. Management decides to add a feature to display where the on-line order was placed from. This information is not in the RDBMS, but in the website logs as the IP address at the time the order was placed. We then get street address through a on-line service which will perform a GeoIP look up.

This now becomes a big data problem because:

- volume: besides the enterprise RDBMS, we now have 2 additional data sources: the web site's log files, and the GeoIP database.
- variety: we have structured data from the DBMS, semi-structured data from the web log files, and data in a 3rd party on-line service.
- velocity: we need to present the information in real-time, processing them from these three sources, and somehow combining them into a single coherent result.

Other factors

Big data visionaries push other factors in big data:

- **veracity**: Veracity addresses the uncertainty of your data. If we want to use data in our analysis, it needs to be trustworthy. This is especially true when dealing with data from external sources and user generated content from social media and the web.
- **viability**: Viability addresses the organizations ability to predict results from data. This is the primary role of the data scientist - to discover significant patterns and predictors in the data. Big data allows us to collect more data points and attributes which may be determining factors of

discovering a predictor 4.

- **value**: Value is the meaning we derive from data. Organizations use value to make decisions; so data data without value is useless5.

What Types of Data are Big Data?

Here are the new types of data being used to compliment business data6:

- **Clickstream** - Analyze website visitors and how they interact with your website for better design and optimization.
- **Sensors** - Capture data from remote sensors to detect patterns in the data stream.
- **Geographic** - Analyze location-based data for insights.
- **Server Logs** - Detect trends and patterns through log activity to detect intrusions or prevent security breaches.
- **Sentiment** - get a sense of your brand through social media and the web.
- **Unstructured** - Emails, documents, photos, audio, video, source code.

Summary

Big Data poses a technological challenge to organizations due to its volume, velocity and variety, as it is no longer possible to store, process and analyze this data with traditional methods. In the next section we explore Hadoop - a software ecosystem designed do address the technological challenges of big data; helping organizations to determine the viability of and acquire value from their data.

Introduction to Hadoop

In this section we will discuss the Hadoop ecosystem - a collection of software components for the storage, processing and analysis of data.

What is Hadoop ?

Hadoop is a highly modular collection of software applications, libraries and API's allowing for distributed storage and processing of large data sets7. The Hadoop ecosystem was designed to run on a single computer but can scale to thousands of computers, called nodes. Regardless of the number of nodes, they appears as a single, transparent system to the end-user and applications, known as a cluster. The Hadoop cluster is designed handle high-availability and fault-tolerance, thus eliminating the need to buy expensive hardware such as RAID storage arrays, redundant network and power supplies.

[4] https://www.linkedin.com/pulse/20140306073407-64875646-big-data-the-5-vs-everyone-must-know

[5] http://www.wired.com/insights/2013/05/the-missing-vs-in-big-data-viability-and-value/

[6] http://info.hortonworks.com/rs/h2source/images/Hadoop-Data-Lake-white-paper.pdf

[7] https://hadoop.apache.org/

These safeguards built into the Hadoop ecosystem mean that hardware failures will not result in downtime or data loss, and that new / replacement nodes can be added trivially.

Why is Something Like This Needed?

The Hadoop we know today originated within the web start-up world at places like Google, Yahoo and Facebook. These companies face challenges dealing with the storage and processing of the massive amount of data generated by their users. In many cases the amount of data generated each day would fill up a best-in class enterprise storage array and would take more than 24 hours to process resulting in a permanent back-log of data activity[8].

Today's Hadoop is not just used by the web giants, but by all sorts of organizations big and small. Every organization has big data, just like it has business data. Organizations of today use Hadoop to deal with high volume, high velocity and unstructured data to discover insights and drive better decision making. And just like the web giants, their data needs to be processed within a reasonable time frame to produce relevant information. The key difference is most organizations don't need a 40,000 node Hadoop cluster to do it[9].

Example

It can be difficult to conceptualize data at such a large scale, so let's explain the concept with a relate-able yet slightly over-simplified example.

Imagine our company has log data of customer activity from our website. We'd like to do something fairly common with this data - get a sense for which percentage of users visit the site with a mobile device and determine if there's a trend of more users accessing it via mobile over time. If the number of mobile users is trending up significantly we'll take action and make an investment in a more responsive design in the future.

Before we start our analysis we hit our first roadblock - one year's worth of log files are 1.2 Terabytes (TB) exceeds the storage capacity of the system where we will perform the analysis. Our work-around strategy is to break up the log files monthly into 100 Megabyte files and process each month, adding the file to the server, running the program then deleting it. Not ideal, but manageable. You write a python script to analyze monthly each file, classifying each request as mobile or non-mobile and tallying counts and hit your second road block. This program takes 3 hours to execute, meaning to analyze the entire set will take 36 hours!

You might argue that this is not a significant issue. We only need to run this program one time for each month and once we have the results the source data and program is no longer required. While that is true, what if the program has errors, and need to be re-run? Analyses like this almost always result in more questions - are they iPhone or Android users? which versions of Android? And these questions mean we need to write a different program, go back to the source data and re-analyze it.

[8] http://www.techrepublic.com/article/why-the-worlds-largest-hadoop-installation-may-soon-become-the-norm/

[9] https://gigaom.com/2012/08/22/facebook-is-collecting-your-data-500-terabytes-a-day/

Let's imagine how Hadoop can help, without explaining how it works in detail, as we'll get to that in the next section. First a Hadoop cluster would automatically distribute our 1.2 Terabytes of logs over each computer node. To us it looks like a single file, which gives us the distinct advantage of executing the program once instead of 12 times. Our program must be modified to understand how to execute over the data on each node, summarizing it and then re-assembling it into a unified result, but in the end we're creating a single program, running over a single data set, and achieving a combined result (instead of 12 results we must combine together). Most importantly, we can automate the process, scheduling this program to run monthly, and write derivations of the program more easily. Finally, we can always "throw more hardware at the problem" to reduce the running time of our program. Adding more nodes to the cluster means less data is being processed at each node, and since this is transparent to us we don't need to modify our program.

How it works

In order to leverage Hadoop effectively, you need to understand how the data processing and data storage frameworks work. You don't need to be a computer engineer, but a high-level understanding of the working parts is essential to building solutions which will work to your advantage.

Architecture

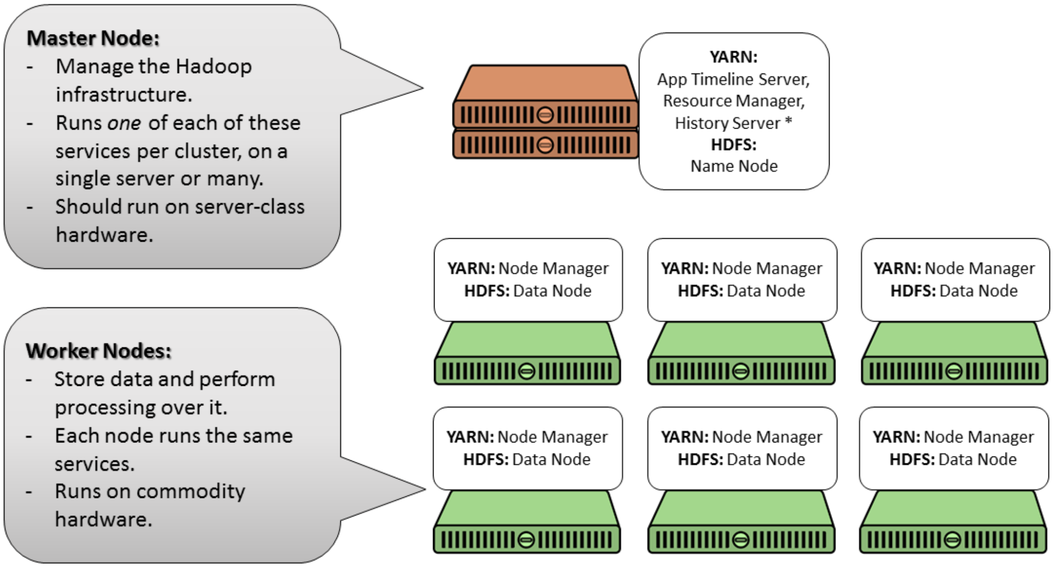
As we've discussed, a Hadoop cluster consists of multiple computers called nodes. Each node has one of two roles.

- **Worker Nodes** are responsible for storage of distributed data and for performing data processing over that data. The typical cluster has as many worker nodes as are required to satisfy the processing and storage needs of the organization. Worker nodes are a commodity and can be added or replaced with minimal affect on the health of the cluster.

- **Master Nodes** are responsible for managing the infrastructure, keeping track on the health of the worker nodes and tracking jobs running over the cluster. The cluster cannot be used without working master nodes and thus in production environments it is essential to configure them to support high-availability scenarios.

The nodes must be able to communicate with each other over a network to function properly. You can build your cluster nodes across the internet, but it is advisable that nodes have have a high throughput network connection between them.

Earlier, we explained that Hadoop is a distributed data storage and data processing framework. Distributed data storage is provided by the **HDFS (Hadoop Distributed File System)** service, and distributed processing is provided by **YARN (Yet Another Resource Negotiator)**. YARN is essentially a distributed operating system that provides a baseline of services for distributed applications.



Master Node:

- Manage the Hadoop infrastructure.
- Runs *one* of each of these services per cluster, on a single server or many.
- Should run on server-class hardware.

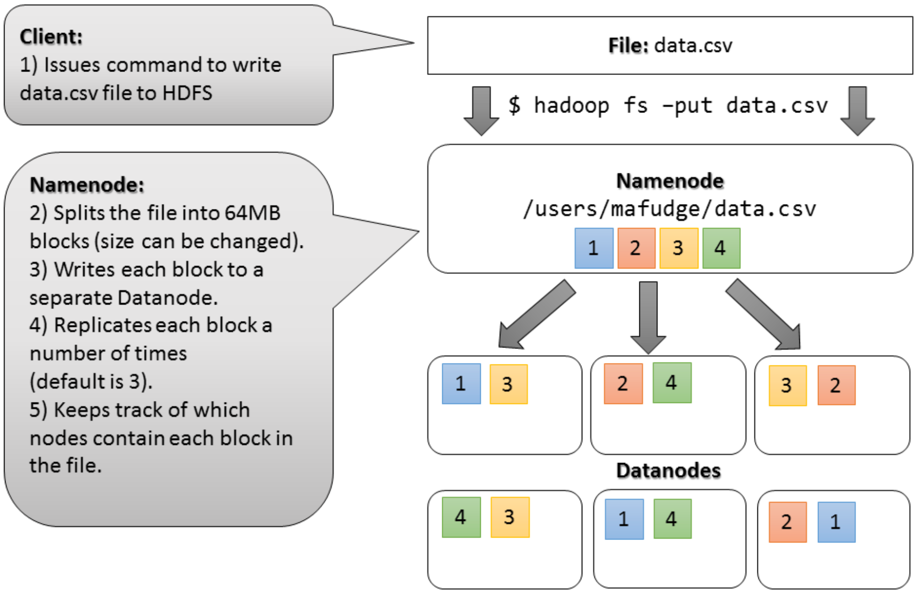
Worker Nodes:

- Store data and perform processing over it.
- Each node runs the same services.
- Runs on commodity hardware.

* Map Reduce 2 service on YARN

HDFS

HDFS consists of a **Namenode** master service and several **DataNode** services - one running on each worker node. The Namenode service is responsible for file-system record keeping. It tracks all the files stored in HDFS and important metadata such as which blocks of the file are stored on which Datanodes. To provide data redundancy, by default each data block within a file is written 3 times - one time each to a different data node. This is a default which can be changed as required.



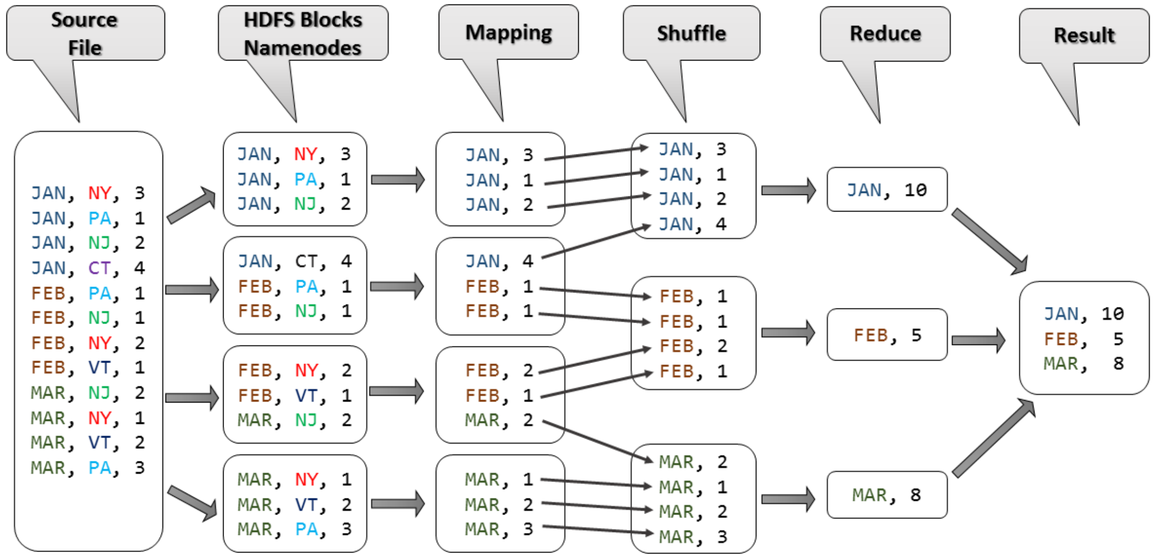
Client:
1) Issues command to write data.csv file to HDFS

Namenode:
2) Splits the file into 64MB blocks (size can be changed).
3) Writes each block to a separate Datanode.
4) Replicates each block a number of times (default is 3).
5) Keeps track of which nodes contain each block in the file.

```
$ hadoop fs -put data.csv
```

Map-Reduce and YARN

At the heart of Hadoop is MapReduce - a distributed data processing framework. MapReduce gets its foundation from pure functional programming languages like LISP. The process consists of two functions: 1) a **mapper** function which transforms items in a list based on a key and value, and a **reducer** function which performs a calculation over the mapped value to achieve a single result for each key. There are two hidden actions in the MapReduce process which are necessary for distributed processing. Before mapped data can be reduced, it needs to be sorted or groups into like keys. This is called a shuffle. After the reduce the outputs need to be combined into a single result of one key and value.



Let's walk through the figure above. In this example we have a data file of sales by Month and State. For example, the first line in the source file says we sold 3 items in New York (NY) in January (JAN). Let's assume we'd like to know how many items were sold each month. Here's the process:

1. The source file is Loaded into HDFS and distributed over the nodes (4 nodes in this example).

2. In this case our **key** is Month (JAN, FEB, MAR, ...) and our value is items sold.

3. Our **mapping function** takes the data on each Namenode as maps the key to the value. It is important to recognize the mapping function operates on each row in this case.

4. After the mapping is complete, the shuffle phase combines keys with the same value onto single nodes, preparing the reducing function.

5. The **reducing function** takes the single key (in this simple case there is one key on three nodes) and sums the items sold for that key, the end product being an single row of key and value. For example, the sum of all the JAN mappings is 3 + 1 + 2 + 4 =10. In the final

6. The reduces function run on the three nodes are combined into a single output, which can be displayed to the console or stored as a file back into HDFS.

With version 2.0, Hadoop was re-engineered to be a multipurpose distributed data processing framework. It can still perform MapReduce, but also provides a framework suited to a variety of other scenarios like data streaming, search, real-time and in-memory distributed data processing, ushering in a new and creative ways to use the ecosystem.

Components of the Hadoop Ecosystem

A primary barrier to learning Hadoop is understanding all of its working parts. Knowing "which system does what" and "which tool should I use to accomplish task X" is something you need to understand before you can begin to leverage Hadoop for your data needs. This section attempts to help you with that at a very high level. It should be noted this is not an exhaustive list, and only covers the essentials for getting started.

Essential Tools

- **HDFS** The distributed file system. Any data you want to process over the Hadoop cluster must be loaded into HDFS.

- **YARN** The data operating system, making it possible to distribute data processing over your cluster. It is the foundation for a variety of popular Hadoop applications like: MapReduce, Hive, and Spark.

- **MapReduce** A batch processing Framework for Hadoop. Programs are typically written in Java, which can be quite cumbersome, and thus there are a variety of other frameworks like **Cascading** to simplify the programming. In addition there are "Compile to Java Virtual Machine" languages like **Scala** and **Py4J**. In addition, you can write MapReduce in an language using the **Hadoop Streaming API**. If you're not a programmer, you can use a friendly scripting language like **Pig** or try to express it in **SQL** using **Hive**.

- **Pig** Pig is a distributed scripting language designed for building and executing data flows. Pig, originally developed by Yahoo!, uses a custom language called **Pig-Latin** which is compiled into a MapReduce program. Should you need to perform custom processing, you can create your own user-defined functions in Java, Scala, or Python. Pig is useful for data cleansing, ETL and general-purpose querying.

- **Hive** Hive, developed by Facebook adds an SQL-like query layer on top of HDFS data. Hive uses a proprietary SQL, called **HiveQL** (Hive Query Language). Similar to SQL, before you can query the data it must reside in a table. Hive has a service **HCatalog**, which manages the metadata: databases, tables, views, functions, etc... HCatalog allows you to easily create tables from existing files in HDFS. Hive's comfortable SQL-like syntax makes it well suited to ad-hoc query and data exploration.

- **Spark** Developed at UC Berkely, Spark is considered the next generation of distributed programming. It is useful for performing ad-hoc analysis of HDFS data, and includes support for a variety of libraries such as data frames (in memory tables), data streaming, machine learning, and graphs, making this platform well suited for a variety of tasks. Spark programs can be written in **Java**, **Scala**, or **Python**.

- **Hue** - HUE stands for Hadoop User Experience. It is a web front-end for HDFS, Hive, HCatalog, Pig and Oozie.
- **JupyterHub** Jupyterhub is a Notebook application, allowing you to write R, Python and Spark programs in the browser, display output and charts, and produce documentation with your code.

Other Tools

- **Flume** Flume is a framework for collecting, aggregating and moving log data. It is commonly used to stream log data into HDFS.
- **Sqoop** Sqoop provides data transfer between HDFS and Relational Database Management Systems. It is useful for getting Relational table data into HDFS, and vice-versa.
- **Kafka** Developed by LinkedIn, Kafka is an Enterprise Service bus / message queuing system.
- **Solr** Solr provides distributed search over HDFS data. It can be used to build a global search mechanism for content in HDFS, or can be used to build scalable search features into external applications.
- **HBase**. HBase, based on Google's BigTable, is a distributed column-oriented database. HBase is useful for storing large quantities of static data which will be aggregated.
- **Storm** Storm is an event processing and distributed processing framework. Spark programs are commonly written in the **Clojure** programming language.
- **Oozie** Oozie is a workflow scheduler for Hadoop. Schedule the execution of jobs based on time or availability of data.

For more information on the Hadoop ecosystem, check out: http://hadoopecosystemtable.github.io/