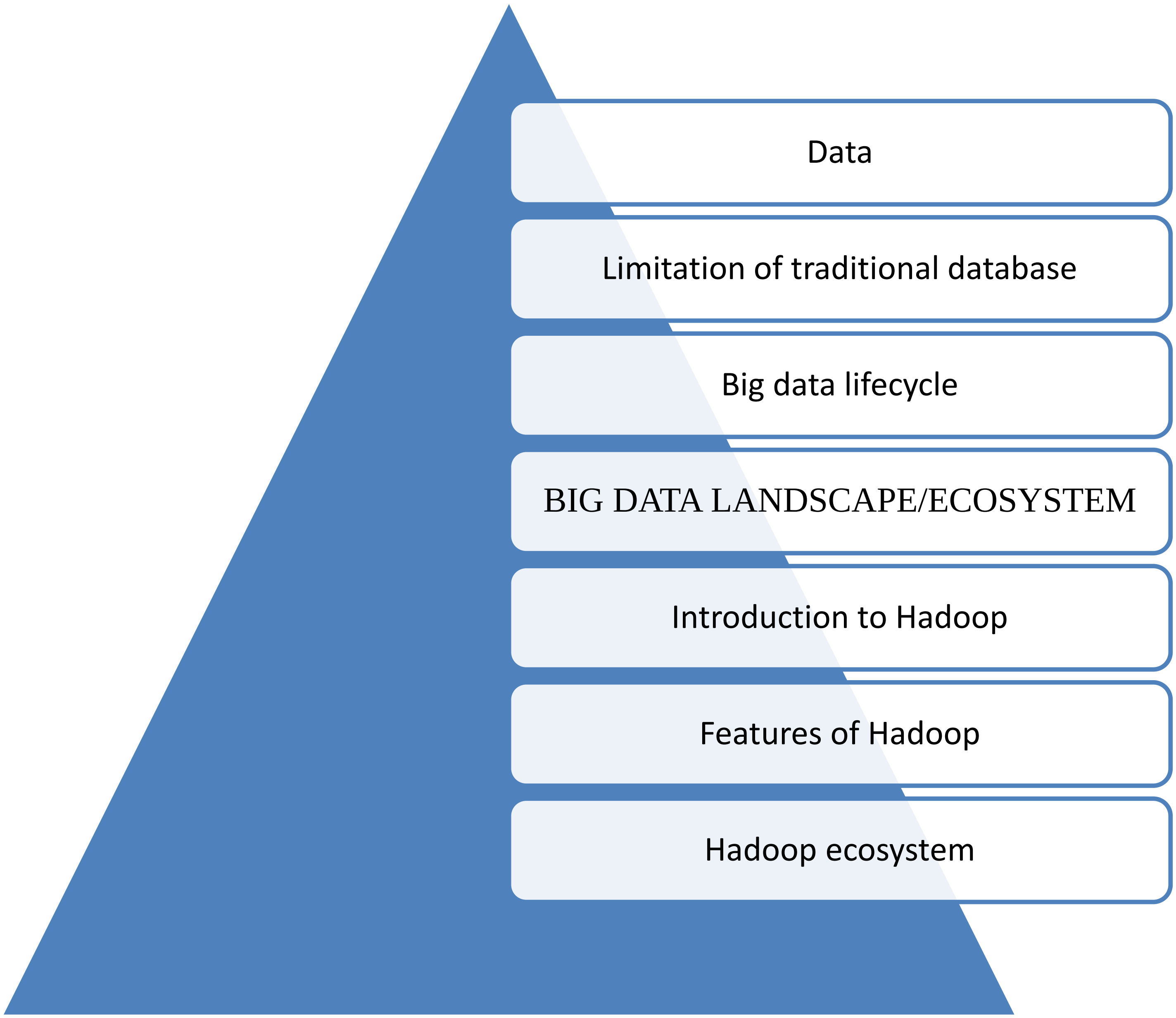
# Introduction to Bigdata and Hadoop

Lecture 2 and lectur3
Dr. S. Srivastava



Data

Limitation of traditional database

Big data lifecycle

BIG DATA LANDSCAPE/ECOSYSTEM

Introduction to Hadoop

Features of Hadoop

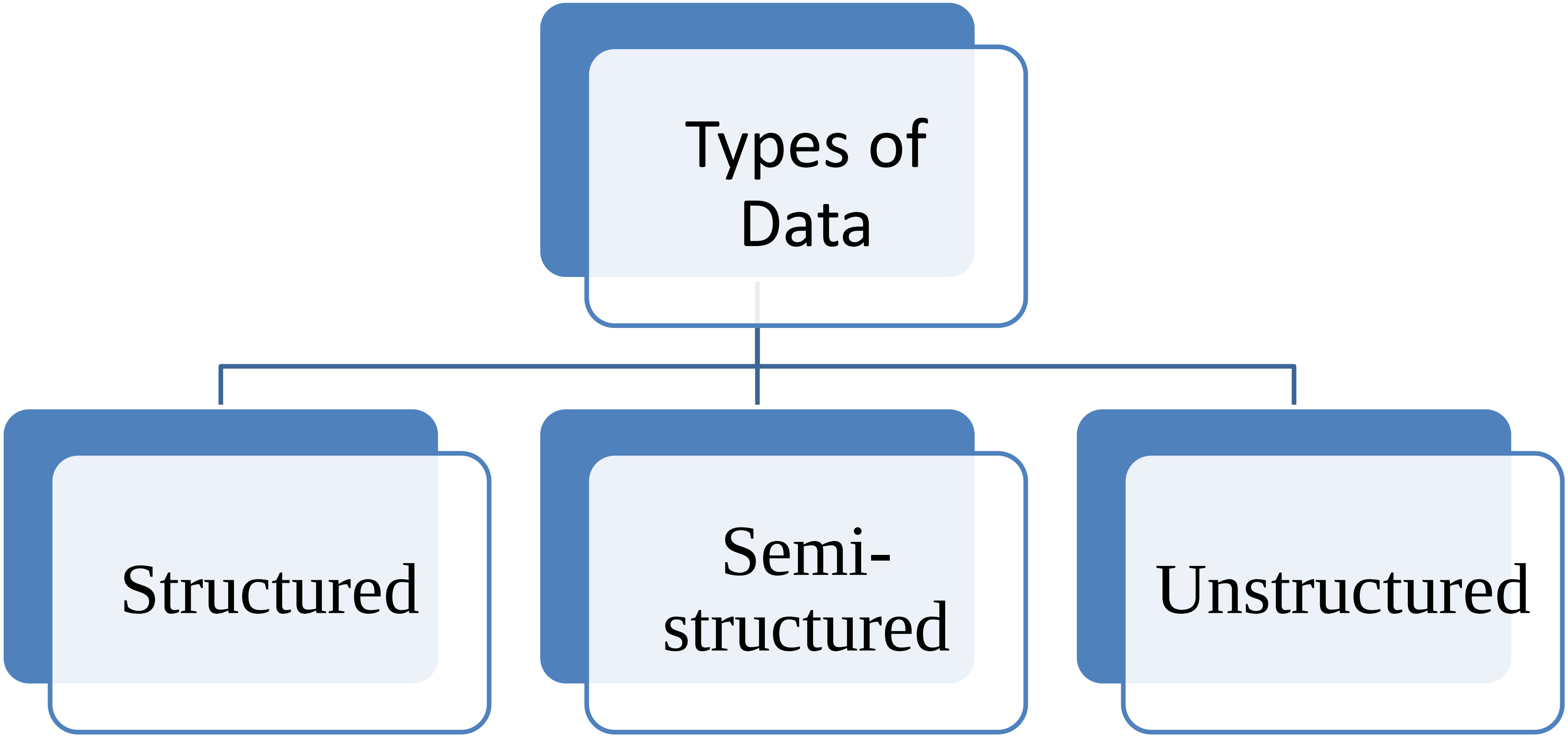
Hadoop ecosystem

## Data

- Data is distinct pieces of information, usually formatted in a special way.

## Examples

- text documents, images, audio clips, software programs, or other types of data.



Types of Data

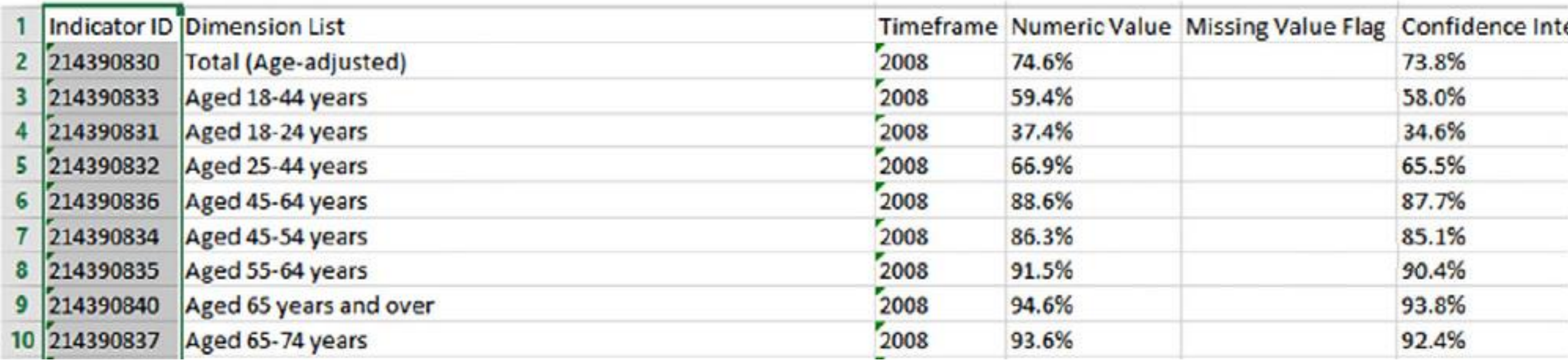
Structured

Semi-structured

Unstructured

# Structured data

- depends on a data model and resides in a fixed field within a record.
- easy to store structured data in tables within databases or Excel files.
- SQL, or Structured Query Language, is the preferred way to manage and query data that resides in databases.



|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| 1 | Indicator ID | Dimension List | Timeframe | Numeric Value | Missing Value Flag | Confidence Inte |
| 2 | 214390830 | Total (Age-adjusted) | 2008 | 74.6% |  | 73.8% |
| 3 | 214390833 | Aged 18-44 years | 2008 | 59.4% |  | 58.0% |
| 4 | 214390831 | Aged 18-24 years | 2008 | 37.4% |  | 34.6% |
| 5 | 214390832 | Aged 25-44 years | 2008 | 66.9% |  | 65.5% |
| 6 | 214390836 | Aged 45-64 years | 2008 | 88.6% |  | 87.7% |
| 7 | 214390834 | Aged 45-54 years | 2008 | 86.3% |  | 85.1% |
| 8 | 214390835 | Aged 55-64 years | 2008 | 91.5% |  | 90.4% |
| 9 | 214390840 | Aged 65 years and over | 2008 | 94.6% |  | 93.8% |
| 10 | 214390837 | Aged 65-74 years | 2008 | 93.6% |  | 92.4% |

## *Unstructured data*

- Not easy to fit into a data model because the content is context-specific or varying.

## Example

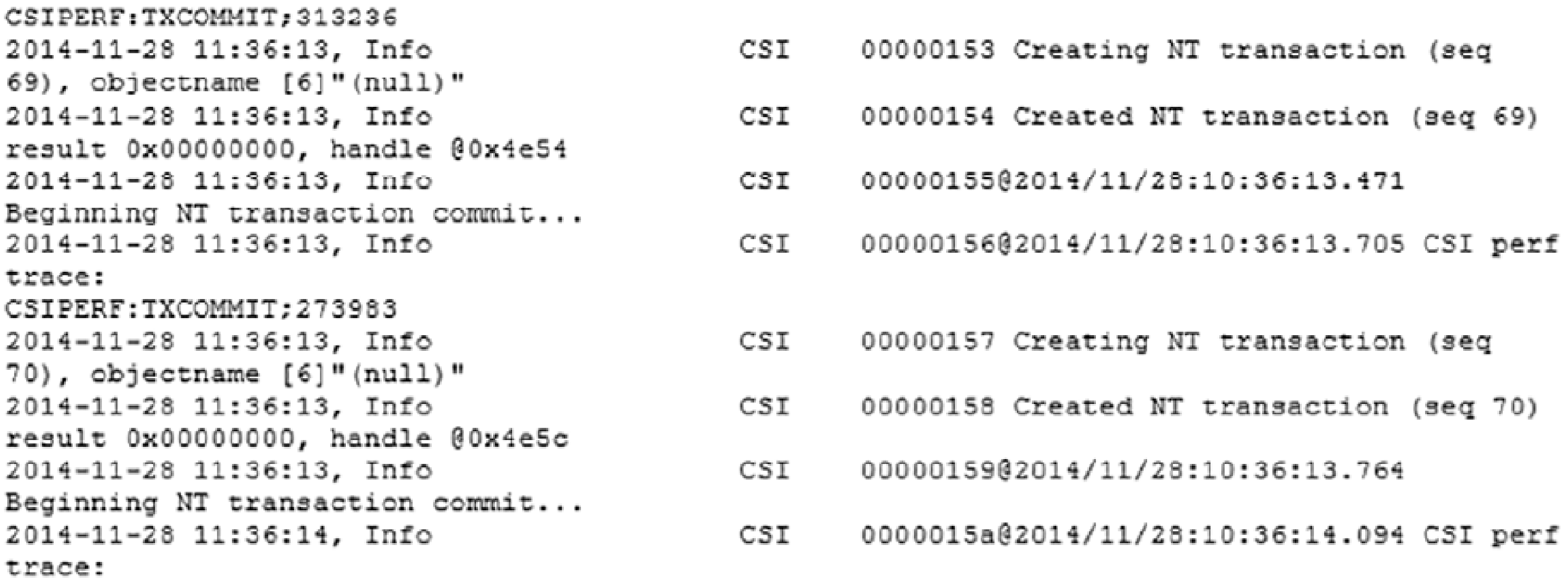
- Although email contains structured elements such as the sender, title, and body text, it’s a challenge to find the number of people who have written an email complaint about a specific employee because so many ways exist to refer to a person, for example.
- The thousands of different languages out there further complicate this

# Example- ***Natural language***

- Natural language is a special type of unstructured data
- it’s challenging to process because it requires knowledge of specific data science techniques and linguistics.
- The natural language processing community has had success in entity recognition, topic recognition, summarization, text completion, and sentiment analysis, but models trained in one domain don’t generalize well to other domains
- Even state-of-the-art techniques aren’t able to decipher the meaning of every piece of text.
- The concept of meaning itself is questionable here. The meaning of the same words can vary when coming from someone upset or joyous.

# Machine-generated data

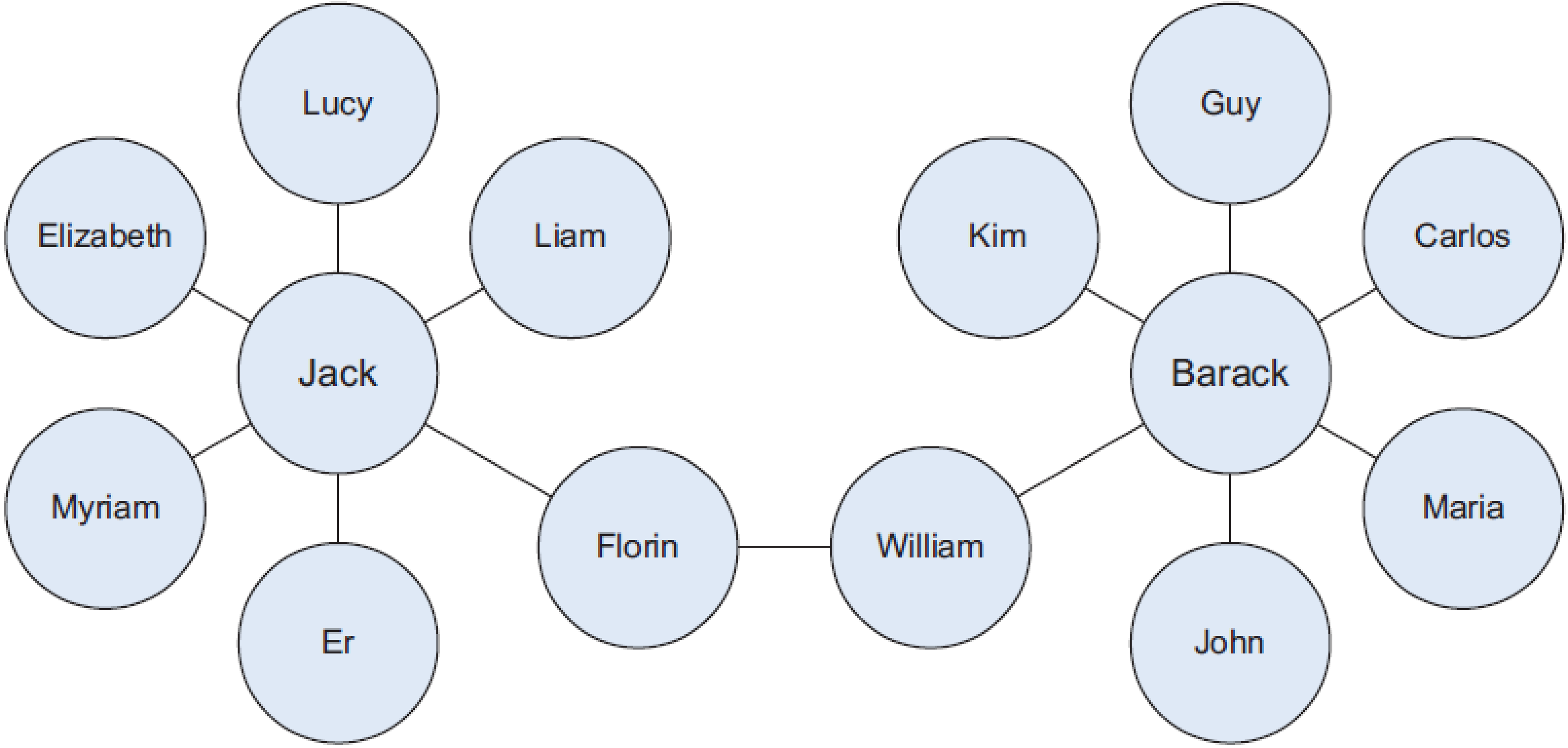
- Machine-generated data is information that's automatically created by a computer, process, application, or other machine without human intervention.
- Machine-generated data is becoming a major data resource and will continue to do so.
- IDC (International Data Corporation)has estimated there will be 26 times more connected things than people in 2020. This network is commonly referred to as *the internet of things*.
- The analysis of machine data relies on highly scalable tools, due to its high volume and speed. Examples of machine data are web server logs, call detail records, network event logs, and telemetry.



```
CSIPERF:TXCOMMIT;313236
2014-11-28 11:36:13, Info                  CSI    00000153 Creating NT transaction (seq
69), objectname [6]"(null)"
2014-11-28 11:36:13, Info                  CSI    00000154 Created NT transaction (seq 69)
result 0x00000000, handle @0x4e54
2014-11-28 11:36:13, Info                  CSI    00000155@2014/11/28:10:36:13.471
Beginning NT transaction commit...
2014-11-28 11:36:13, Info                  CSI    00000156@2014/11/28:10:36:13.705 CSI perf
trace:
CSIPERF:TXCOMMIT;273983
2014-11-28 11:36:13, Info                  CSI    00000157 Creating NT transaction (seq
70), objectname [6]"(null)"
2014-11-28 11:36:13, Info                  CSI    00000158 Created NT transaction (seq 70)
result 0x00000000, handle @0x4e5c
2014-11-28 11:36:13, Info                  CSI    00000159@2014/11/28:10:36:13.764
Beginning NT transaction commit...
2014-11-28 11:36:14, Info                  CSI    0000015a@2014/11/28:10:36:14.094 CSI perf
trace:
```

# *Graph-based or network data*

- *A graph is a* mathematical structure to model pair-wise relationships between objects
- The graph structures use nodes, edges, and properties to represent and store graphical data.
- Graph-based data is a natural way to represent social networks, and its structure allows you to calculate specific metrics such as the influence of a person and the shortest path between two people
- Examples of graph-based data can be found on many social media websites.
- For instance, on LinkedIn you can see who you know at which company.Your follower list on Twitter is another example of graph-based data.
- The connecting edges here to show “friends” on Facebook. Imagine another graph with the same people which connects business colleagues via LinkedIn. Imagine a third graph based on movie interests on Netflix.
- Overlapping the three different-looking graphs makes more interesting questions possible.



**Friends in a social network are an example of graph-based data.**

# *Audio, image, and video*

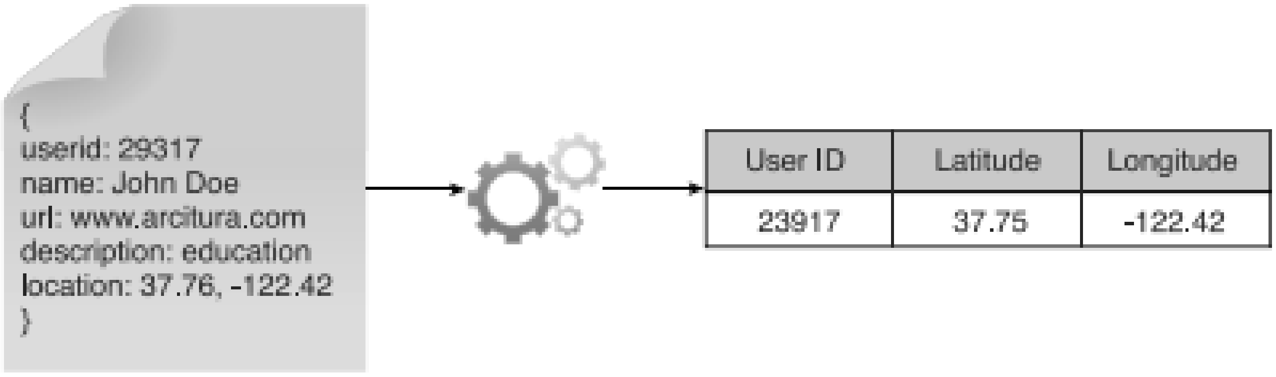
- Audio, image, and video are data types that pose specific challenges to a data scientist.
- Tasks that are trivial for humans, such as recognizing objects in pictures, turn out to be challenging for computers.
- MLBAM (Major League Baseball Advanced Media) announced in 2014 that they'll increase video capture to approximately 7 TB per game for the purpose of live, in-game analytics.
- High-speed cameras at stadiums will capture ball and athlete movements to calculate in real time, for example, the path taken by a defender relative to two baselines.
- Recently a company called DeepMind succeeded at creating an algorithm that's capable of learning how to play video games.
- This algorithm takes the video screen as input and learns to interpret everything via a complex process of deep learning.

# *Streaming data*

- While streaming data can take almost any of the previous forms, it has an extra property.
- The data flows into the system when an event happens instead of being loaded into a data store in a batch.
- Examples are the “What’s trending” on Twitter, live sporting or music events, and the stock market.

# Semi-structured Data

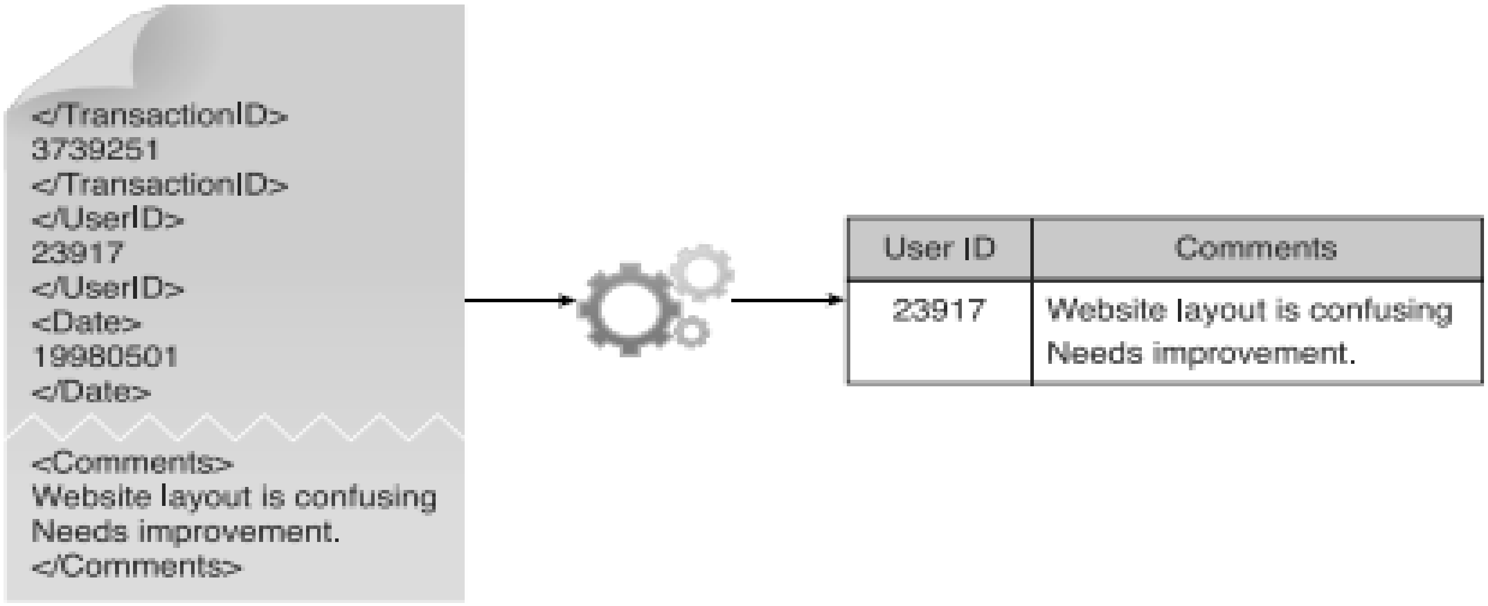
- It has a defined level of structure and consistency but not relational in nature.
- This kind of data is commonly stored in files that contain text.
- Eg- JSON and XML files
- Due to textual nature of this data and its conformance to some level of structure, it is more easily processed than unstructured data



```
{
userid: 29317
name: John Doe
url: www.arcitura.com
description: education
location: 37.76, -122.42
}
```

| User ID | Latitude | Longitude |
|---|---|---|
| 23917 | 37.75 | -122.42 |

The user ID and coordinates of a user are extracted from a single JSON field.



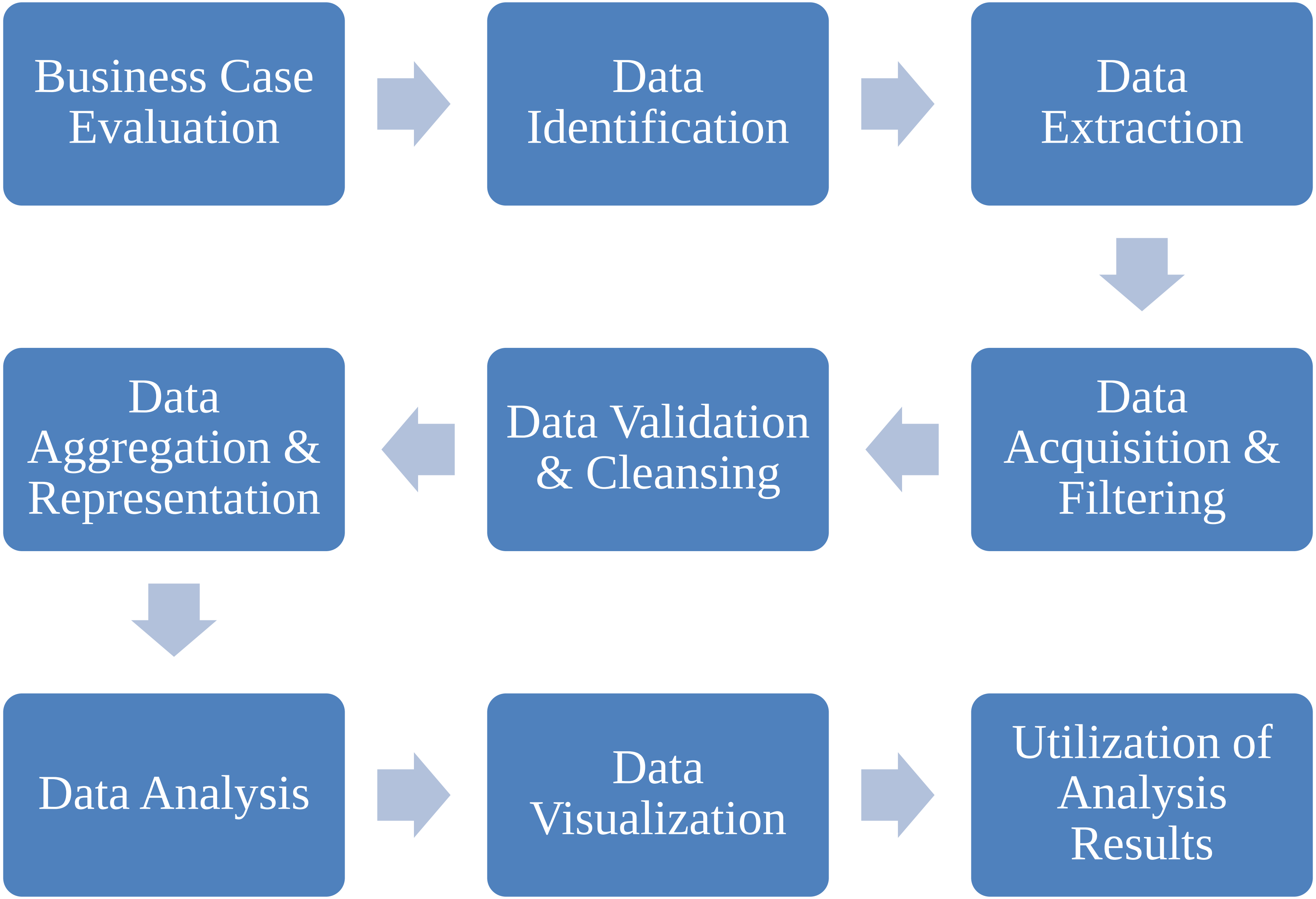
```
</TransactionID>
3739251
</TransactionID>
</UserID>
23917
</UserID>
<Date>
19980501
</Date>
<Comments>
Website layout is confusing
Needs improvement.
</Comments>
```

| User ID | Comments |
|---|---|
| 23917 | Website layout is confusing Needs improvement. |

Comments and user IDs are extracted from an XML document.

# Big Data Analytics Lifecycle

- Big Data analysis differs from traditional data analysis primarily due to the volume, velocity and variety characteristics of the data being processes.
- To address the distinct requirements for performing analysis on Big Data, a step-by-step methodology is needed to organize the activities and tasks involved with acquiring, processing, analyzing and repurposing data



Business Case Evaluation

Data Identification

Data Extraction

Data Acquisition & Filtering

Data Validation & Cleansing

Data Aggregation & Representation

Data Analysis

Data Visualization

Utilization of Analysis Results

Nine stages of Big Data analytics lifecycle

# The Business Case Evaluation

- begin with a well-defined business case that presents a clear understanding of the justification, motivation and goals of carrying out the analysis.
- The Business Case Evaluation stage requires that a business case be created, assessed and approved prior to proceeding with the actual hands-on analysis tasks.
- helps decision-makers understand the business resources that will need to be utilized and which business challenges the analysis will tackle.
- identification of KPIs (key performance indicator) during this stage can help determine assessment criteria and guidance for the evaluation of the analytic results.

- If KPIs are not readily available, efforts should be made to make the goals of the analysis project SMART, which stands for specific, measurable, attainable, relevant and timely.

- Based on business requirements that are documented in the business case, it can be determined whether the business problems being addressed are really Big Data problems.

- In order to qualify as a Big Data problem, a business problem needs to be directly related to one or more of the Big Data characteristics of volume, velocity, or variety.

- Another outcome of this stage is the determination of the underlying budget required to carry out the analysis project.

- Any required purchase, such as tools, hardware and training, must be understood in advance so that the anticipated investment can be weighed against the expected benefits of achieving the goals.

# Data Identification

- The Data Identification stage is dedicated to identifying the datasets required for the analysis project and their sources.
- Identifying a wider variety of data sources may increase the probability of finding hidden patterns and correlations.
- In the case of internal datasets, a list of available datasets from internal sources, such as data marts and operational systems, are typically compiled and matched against a pre-defined dataset specification.
- In the case of external datasets, a list of possible third-party data providers, such as data markets and publicly available datasets, are compiled.

# Data Acquisition and Filtering

- the data is gathered from all of the data sources that were identified during the previous stage.
- The acquired data is then subjected to automated filtering for the removal of corrupt data or data that has been deemed to have no value to the analysis objectives.
- Depending on the type of data source, data may come as a collection of files, such as data purchased from a third-party data provider, or may require API integration (Application Program Interface), such as with Twitter.
- In many cases, especially where external, unstructured data is concerned, some or most of the acquired data may be irrelevant (noise) and can be discarded as part of the filtering process.

- Data classified as “corrupt” can include records with missing or nonsensical values or invalid data types.

- Data that is filtered out for one analysis may possibly be valuable for a different type of analysis. Therefore, it is advisable to store a verbatim copy of the original dataset before proceeding with the filtering.

- To minimize the required storage space, the verbatim copy can be compressed.

# Data Extraction

- The Data Extraction lifecycle stage, is dedicated to extracting disparate data and transforming it into a format that the underlying Big Data solution can use for the purpose of the data analysis.
- The extent of extraction and transformation required depends on the types of analytics and capabilities of the Big Data solution.
- Example- extracting text for text analytics, which requires scans of whole documents, is simplified if the underlying Big Data solution can directly read the document in its native format.

# Data Validation and Cleansing

- Invalid data can skew and falsify analysis results.
- Unlike traditional enterprise data, where the data structure is pre-defined and data is pre-validated, data input into Big Data analyses can be unstructured without any indication of validity.
- Its complexity can further make it difficult to arrive at a set of suitable validation constraints.
- The Data Validation and Cleansing stage is dedicated to establishing often complex validation rules and removing any known invalid data.
- Big Data solutions often receive redundant data across different datasets. This redundancy can be exploited to explore interconnected datasets in order to assemble validation parameters and fill in missing valid data.

- For example, The first value in Dataset B is validated against its corresponding value in Dataset A.

- The second value in Dataset B is not validated against its corresponding value in Dataset A.

- If a value is missing, it is inserted from Dataset A.

- For real-time analytics, a more complex in-memory system is required to validate and cleanse the data as it arrives from the source

- Data source can play an important role in determining the accuracy and quality of questionable data. Data that appears to be invalid may still be valuable in that it may possess hidden patterns and trends.

# Data Aggregation and Representation

- Data may be spread across multiple datasets, requiring that datasets be joined together via common fields, for example date or ID.
- In other cases, the same data fields may appear in multiple datasets, such as date of birth.
- The Data Aggregation and Representation stage, is dedicated to integrating multiple datasets together to arrive at a unified view.

- Performing this stage can become complicated because of differences in:
- Data Structure – Although the data format may be the same, the data model may be different.
- Semantics – A value that is labelled differently in two different datasets may mean the same thing, for example “surname” and “last name.”

# Data Analysis

- The Data Analysis stage is dedicated to carrying out the actual analysis task, which typically involves one or more types of analytics.

- This stage can be iterative in nature, especially if the data analysis is exploratory, in which case analysis is repeated until the appropriate pattern or correlation is uncovered.
- The exploratory analysis approach will be explained shortly, along with confirmatory analysis.

- Depending on the type of analytic result required, this stage can be as simple as querying a dataset to compute an aggregation for comparison.

- On the other hand, it can be as challenging as combining data mining and complex statistical analysis techniques to discover patterns and anomalies or to generate a statistical or mathematical model to depict relationships between variables.

- Data analysis can be classified as confirmatory analysis or exploratory analysis , the latter of which is linked to data mining.

- Confirmatory data analysis is a deductive approach where the cause of the phenomenon being investigated is proposed beforehand. The proposed cause or assumption is called a hypothesis.

- The data is then analyzed to prove or disprove the hypothesis and provide definitive answers to specific questions. Data sampling techiniques are typically used.

# Data Visualization

- The ability to analyse massive amounts of data and find useful insights carries little value if the only ones that can interpret the results are the analysts.
- The Data Visualization stage, is dedicated to using data visualization techniques and tools to graphically communicate the analysis results for effective interpretation by business users.

- Business users need to be able to understand the results in order to obtain value from the analysis and subsequently have the ability to provide feedback

- The results of completing the Data Visualization stage provide users with the ability to perform visual analysis, allowing for the discovery of answers to questions that users have not yet even formulated.

- The same results may be presented in a number of different ways, which can influence the interpretation of the results.

- Consequently, it is important to use the most suitable visualization technique by keeping the business domain in context.

# Utilization of Analysis Results

- Subsequent to analysis results being made available to business users to support business decision-making, such as via dashboards, there may be further opportunities to utilize the analysis results.

- The Utilization of Analysis Results stage, is dedicated to determining how and where processed analysis data can be further leveraged.

- Depending on the nature of the analysis problems being addressed, it is possible for the analysis results to produce “models” that encapsulate new insights and understandings about the nature of the patterns and relationships that exist within the data that was analyzed.

- A model may look like a mathematical equation or a set of rules. Models can be used to improve business process logic and application system logic, and they can form the basis of a new system or software program.

# BIG DATA LANDSCAPE/ECOSYSTEM

## The big data landscape consists of many different technologies that can be categorized into the following:

- Distributed File system
- Distributed programming frameworks
- Data integration
- Databases
- Machine learning
- Security
- Scheduling
- Benchmarking
- System deployment
- Service programming

# *Distributed file systems*

- A distributed file system is similar to a normal file system, except that it runs on multiple servers at once.
- Because it’s a file system, you can do almost all the same things you’d do on a normal file system.
- Actions such as storing, reading, and deleting files and adding security to files are at the core of every file system, including the distributed one.
- The best-known distributed file system at this moment is the *Hadoop File System (HDFS). It is an open source implementation of the Google File System.*
- However, many other distributed file systems exist: *Red Hat Cluster File System*, *Ceph File System*, and *Tachyon File System, to name but three.*

Distributed file systems have significant advantages:

■ They can store files larger than any one computer disk.

■ Files get automatically replicated across multiple servers for redundancy or parallel operations while hiding the complexity of doing so from the user.

■ The system scales easily: you’re no longer bound by the memory or storage restrictions of a single server.

# *Distributed programming framework*

- Once you have the data stored on the distributed file system, you want to exploit it.
- One important aspect of working on a distributed hard disk is that you won’t move your data to your program, but rather you’ll move your program to the data.
- The open source community has developed many frameworks to handle this for you, and these give you a much better experience working with distributed data and dealing with many of the challenges it carries.

# *Data integration framework*

- Once you have a distributed file system in place, you need to add data.
- You need to move data from one source to another, and this is where the data integration frameworks such as Apache Sqoop and Apache Flume excel.
- The process is similar to an extract, transform, and load process in a traditional data warehouse.

# *Machine learning frameworks*

- When you have the data in place, it’s time to extract the coveted insights. This is where you rely on the fields of machine learning, statistics, and applied mathematics.
- Before World War II everything needed to be calculated by hand, which severely limited the possibilities of data analysis.
- After World War II computers and scientific computing were developed.
- A single computer could do all the counting and calculations and a world of opportunities opened. Ever since this breakthrough, people only need to derive the mathematical formulas, write them in an algorithm, and load their data.
- With the enormous amount of data available nowadays, one computer can no longer handle the workload by itself.

- One of the biggest issues with the old algorithms is that they don’t scale well.

- With the amount of data we need to analyze today, this becomes problematic, and specialized frameworks and libraries are required to deal with this amount of data.

- The most popular machine-learning library for Python is Scikit-learn. It’s a great machine-learning toolbox,

- *PyBrain for neural networks*—Neural networks are learning algorithms that mimic the human brain in learning mechanics and complexity. Neural networks are often regarded as advanced and black box.
- *NLTK or Natural Language Toolkit—its focus is working* with natural language. It’s an extensive library that comes bundled with a number of text corpuses to help you model your own data.
- ■ *Pylearn2—Another machine learning toolbox but a bit less mature than Scikit-learn.*
- ■ *TensorFlow—A Python library for deep learning provided by Google.*

- Spark is a new Apache licensed machine-learning engine, specializing in real-learn-time machine learning.

# *NoSQL databases*

- If you need to store huge amounts of data, you require software that's specialized in managing and querying this data.
- Traditionally this has been the playing field of relational databases such as Oracle SQL, MySQL, Sybase IQ, and others.
- While they're still the go-to technology for many use cases, new types of databases have emerged under the grouping of NoSQL databases.
- Traditional databases had shortcomings that didn't allow them to scale well.
- By solving several of the problems of traditional databases, NoSQL databases allow for a virtually endless growth of data.
- These shortcomings relate to every property of big data: their storage or processing power can't scale beyond a single node and they have no way to handle streaming, graph, or unstructured forms of data.

Many different types of databases have arisen, but they can be categorized into the following types:

■ *Column databases—Data is stored in columns, which allows algorithms to perform* much faster queries.

■ *Document stores—Document stores no longer use tables, but store every observation* in a document. This allows for a much more flexible data scheme.

■ *Streaming data—Data is collected, transformed, and aggregated not in batches* but in real time.

■ *Key*-*value stores—Data isn't stored in a table; rather you assign a key for every* value, such as org.marketing.sales.2015: 20000. This scales well but places almost all the implementation on the developer.

■ *SQL on Hadoop—Batch queries on Hadoop are in a SQL-like language that uses* the map-reduce framework in the background.

# Scheduling tools

- Scheduling tools help you automate repetitive tasks and trigger jobs based on events such as adding a new file to a folder.

- These are similar to tools such as CRON on Linux but are specifically developed for big data. You can use them, for instance, to start a MapReduce task whenever a new dataset is available in a directory.

# *Benchmarking tools*

- This class of tools was developed to optimize your big data installation by providing standardized profiling suites.
- A profiling suite is taken from a representative set of big data jobs.
- Benchmarking and optimizing the big data infrastructure and configuration aren’t often jobs for data scientists themselves but for a professional specialized in setting up IT infrastructure
- Using an optimized infrastructure can make a big cost difference.
- For example, if you can gain 10% on a cluster of 100 servers, you save the cost of 10 servers.

# *System deployment*

- Setting up a big data infrastructure isn’t an easy task and assisting engineers in deploying new applications into the big data cluster is where system deployment tools shine.
- They largely automate the installation and configuration of big data components.
- This isn’t a core task of a data scientist.

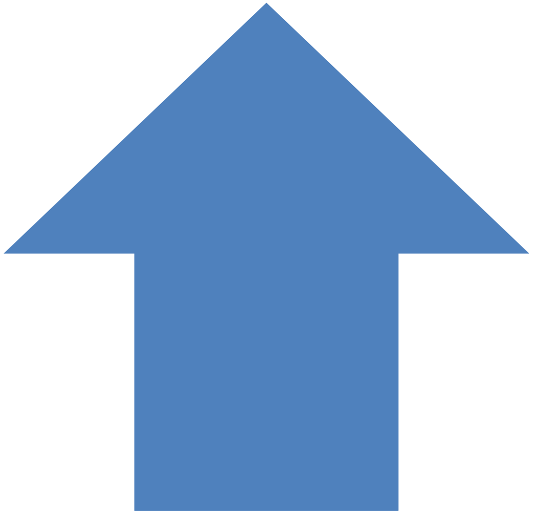
# *Service programming*

- Suppose that you’ve made a world-class soccer prediction application on Hadoop, and you want to allow others to use the predictions made by your application. However, you have no idea of the architecture or technology of everyone keen on using your predictions.
- Service tools excel here by exposing big data applications to other applications as a service.
- Data scientists sometimes need to expose their models through services. The best-known example is the REST service; REST stands for representational state transfer.
- It’s often used to feed websites with data.

# Security

- Do you want everybody to have access to all of your data?
- You probably need to have fine-grained control over the access to data but don’t want to manage this on an application-by-application basis.
- Big data security tools allow you to have central and fine-grained control over access to the data.
- Big data security has become a topic in its own right, and data scientists are usually only confronted with it as data consumers; seldom will they implement the security themselves.

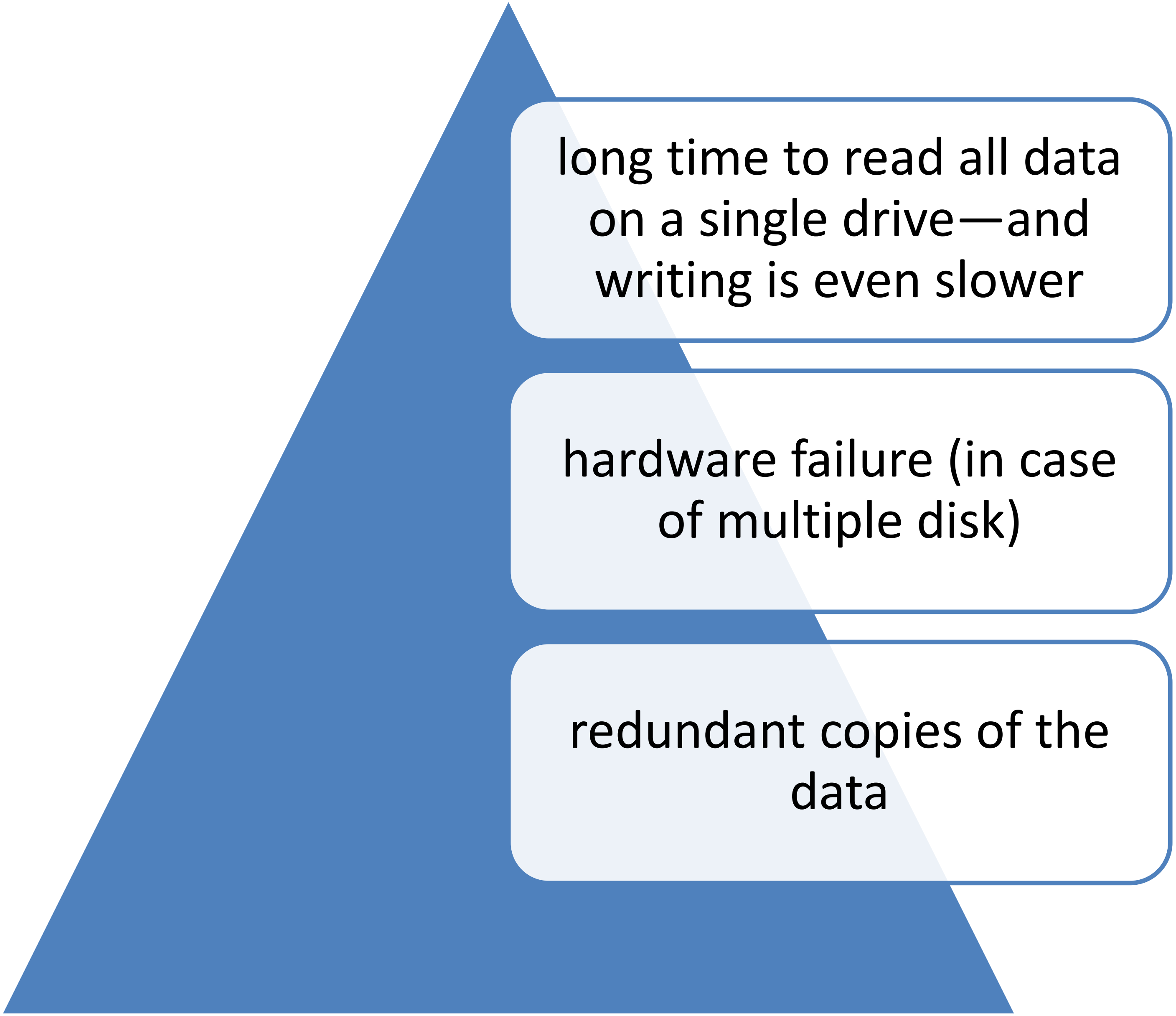
# INTRODUCTION to HADOOP



capacities of hard drives have increased massively over the years

access speeds—the rate at which data can be read from drives have not kept up.

Issues with large data set



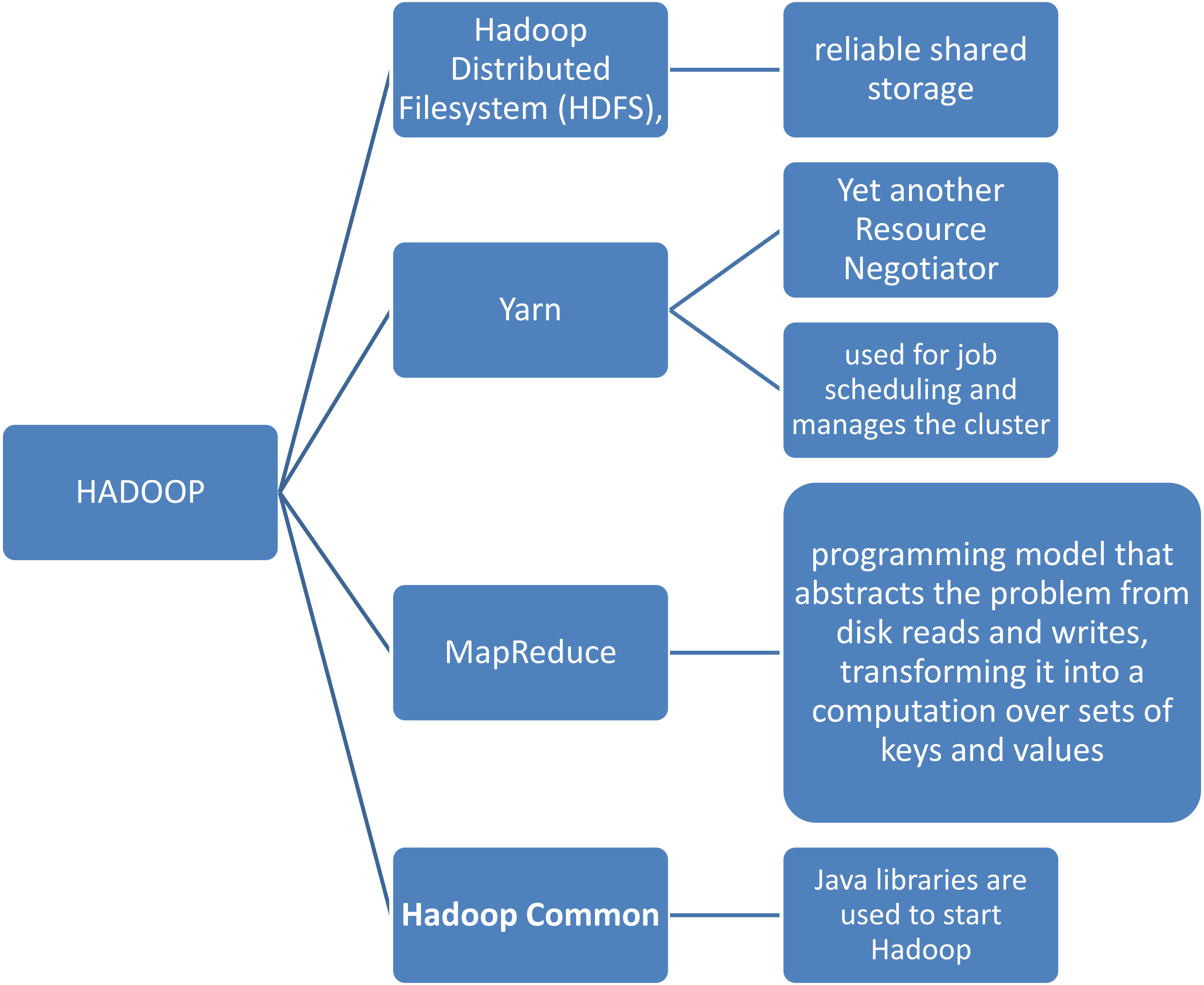
long time to read all data on a single drive—and writing is even slower

hardware failure (in case of multiple disk)

redundant copies of the data

# What is Hadoop?

- Hadoop is fundamentally an open source infrastructure software framework that allows distributed storage and processing a huge amount of data i.e. Big Data.
- It’s a cluster system which works as a Master-Slave Architecture.
- With such architecture, large data can be stored and processed in parallel.
- Different types of data can be analyzed, structured(tables), unstructured (logs, email body, blog text) and semi-structured (media file metadata, XML, HTML).



HADOOP

Hadoop Distributed Filesystem (HDFS),

reliable shared storage

Yarn

Yet another Resource Negotiator

used for job scheduling and manages the cluster

MapReduce

programming model that abstracts the problem from disk reads and writes, transforming it into a computation over sets of keys and values

**Hadoop Common**

Java libraries are used to start Hadoop

# HADOOP VS TRADITIONAL DATABASE

# #1. Data Variety

| Hadoop | RDBMS |
|---|---|
| Used for Structured, Semi Structured and Unstructured data. | Mainly for Structured data. |

# #2. Data Storage

## Hadoop

Use for large data set (Tbs and Pbs).

## RDBMS

Average size data (Gbs).

# #3. Querying

## Hadoop

HQL (Hive Query Language).

## RDBMS

SQL Language.

# #4. Schema

## Hadoop

Required on read (dynamic schema).

## RDBMS

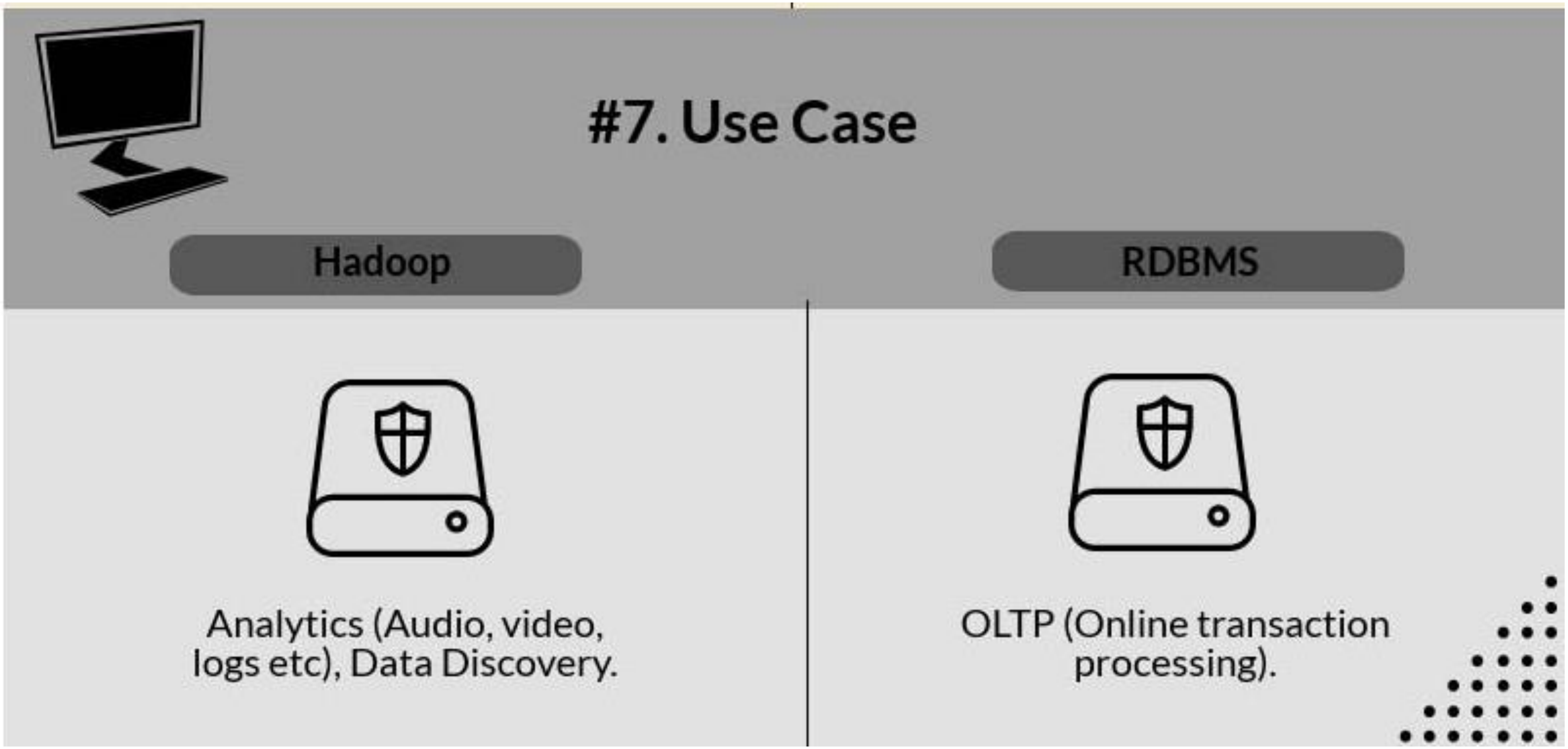
Required on write (static schema).

# #5. Speed

| Hadoop | RDBMS |
|---|---|
| Both read and writes are fast. | Reads are fast. |

# #6. Cost

| Hadoop | RDBMS |
|---|---|
| Free . | License . |



# #7. Use Case

| Hadoop | RDBMS |
|---|---|
| Analytics (Audio, video, logs etc), Data Discovery. | OLTP (Online transaction processing). |

# #8. Data Objects

| Hadoop | RDBMS |
|---|---|
| Works on Key/Value Pair. | Works on Relational Tables. |

# #9. Throughput

| Hadoop | RDBMS |
|---|---|
| High. | Low. |

# #10. Scalability

| Hadoop | RDBMS |
|---|---|
| Horizontal. | Vertical. |

# #11. Hardware Profile

| Hadoop | RDBMS |
|---|---|
| Commodity/Utility Hardware. | High End Servers. |

# #12. Integrity

| Hadoop | RDBMS |
|---|---|
| Low. | High (ACID). |

# Advantages of Hadoop

## Scalable

- it can store and distribute very large data sets across hundreds of systems/servers that operate in parallel.
- Hadoop enables businesses to run applications on thousands of nodes involving thousands of terabytes of data processing.
- And also supports hardware horizontal scalability which can add the nodes during the processing without system downtime.

## Cost effective

- Hadoop offers a cost-effective storage solution for businesses exploding data sets.

## Flexible

- Manages data whether structured or unstructured, encoded or formatted, or any other type of data.
- Hadoop brings the value to the table where unstructured data can be useful in decision making process

## Faster

- Hadoop’s unique storage method is based on a distributed file system.
- The tools for data processing are often on the same servers where the data is located
- If dealing with large volumes of unstructured data, Hadoop is able to efficiently process terabytes of data in just minutes, and petabytes in hours.

## Fault Tolerant

- The data sent to one individual node and the same data also replicates on other nodes in the same cluster.
- If the individual node failed to process the data, the other nodes in the same cluster available to process the data

Disadvantages of Hadoop:

## Security Concerns

- Hadoop security model is disabled by default due to sheer complexity and also missing encryption at the storage and network levels.

## Vulnerable By Nature

- The framework is written almost entirely in Java which is the most widely used controversial programming languages in existence.. Java has been heavily exploited and as a result, implicated in numerous security breaches

## Not Fit for Small Data

- Due to its high capacity design, the HDFS, lacks the ability to efficiently support the reading of small files.

Hadoop features

## Open Source

- Hadoop is an open source project and its code can be modified according to business requirements.

## Distributed Processing

- As data is stored in a distributed manner in HDFS across the cluster and data is processed in parallel on a cluster of nodes.

## Faster

- Hadoop is extremely good at high-volume batch processing because of its ability to do parallel processing.
- Hadoop can perform batch processes multiple times faster than on single thread server or on the mainframe.

## Fault Tolerance

- The data sent to one individual node and the same data also replicates on other nodes in the same cluster.
- If the individual node failed to process the data, the other nodes in the same cluster available to process the data.

## Reliability

- Due to data replication in the cluster, data is reliably stored on the cluster of machine despite machine failures.
- If the node failed to process the data, the data will be stored reliably due to this characteristic of Hadoop.

## High Availability

- Data is highly available and accessible despite hardware failure due to multiple copies of data.
- If the machine or hardware crashes, then data will be accessed from another path.

## Scalability

- Hadoop is a highly scalable storage platform as it can store and distribute very large data sets across hundreds of systems/servers that operate in parallel.
- Hadoop enables businesses to run applications on thousands of nodes involving thousands of terabytes of data processing.

## Flexibility

- Hadoop manages data whether structured or unstructured, encoded or formatted, or any other type of data.
- Businesses can use Hadoop to derive valuable business insights from data sources such as social media, email conversations.
- Hadoop brings the value to the table where unstructured data can be useful in decision making process

## Economic/Cost effective

- Hadoop offers a cost-effective storage solution for businesses exploding data sets.
- Hadoop is not very expensive as it runs on a cluster of commodity hardware.

## Easy to use

- No need of client to deal with distributed computing, the framework takes care of all the things. So, Hadoop is easy to use.

## Data Locality

- This one is a unique feature of Hadoop that made it easily handle the Big Data. When a client submits the MapReduce algorithm, this algorithm is moved to data in the cluster rather than bringing data to the location where the algorithm is submitted and then processing it.

# Hadoop - Architecture

Hadoop efficiently stores large volumes of data on a cluster of commodity hardware.

Hadoop not only a storage system but also platform for processing large data along with storage.

There are mainly five building blocks inside this runtime environment

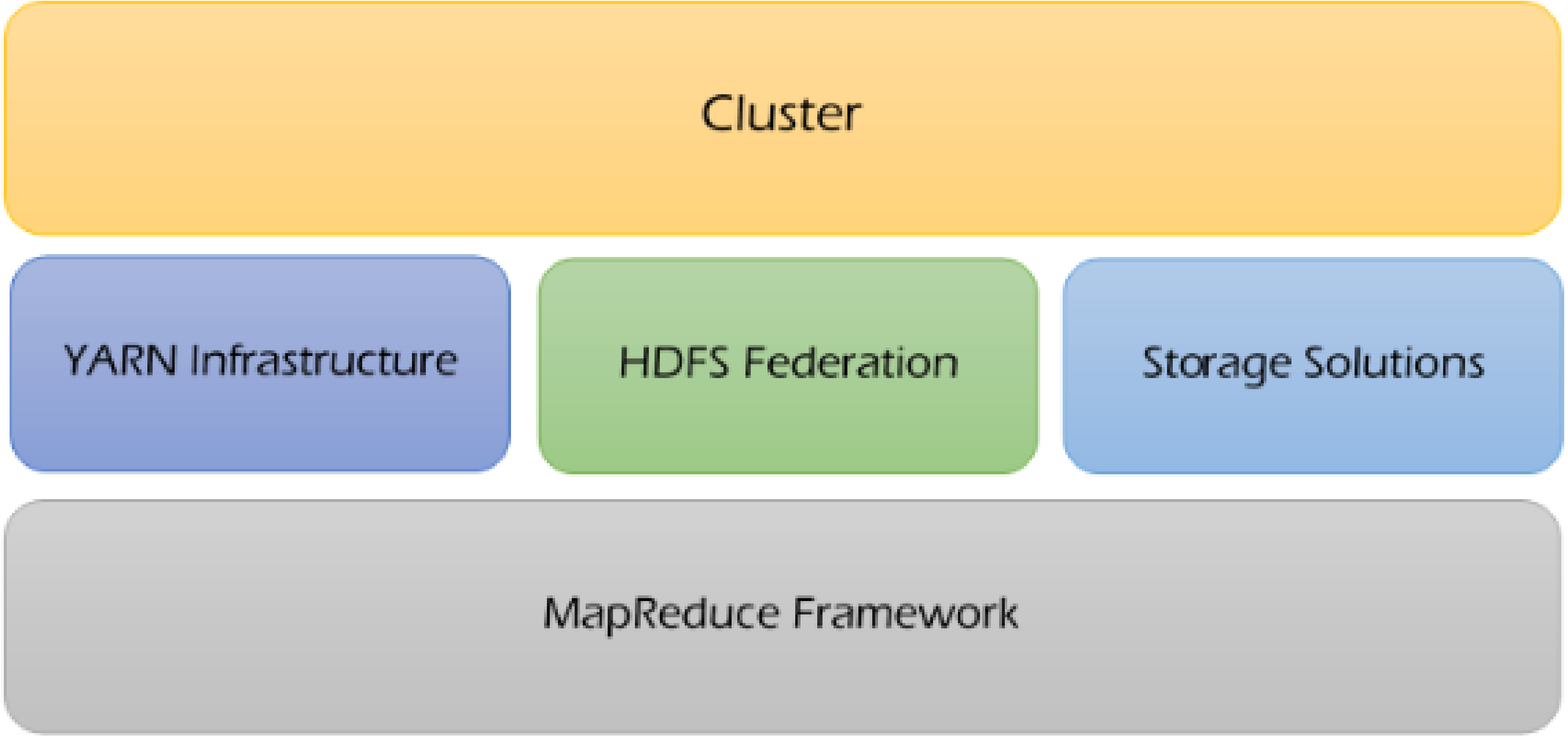


Fig.  building blocks of hadoop

# Hadoop Cluster: -

- Designed specifically for storing and analyzing huge amounts of unstructured data in a distributed computing environment.
- Cluster is the set of nodes which are also known as host machines.
- Cluster is the hardware part of the infrastructure.
- Hadoop clusters are often referred to as "shared nothing" systems because the only thing that is shared between nodes is the network that connects them.
- Known for boosting the data analysis applications speed.
- Hadoop clusters are highly scalable: If a cluster's processing data growing by volumes, new additional cluster nodes can be added to increase throughput.
- Hadoop clusters are highly resistant to failure because the data always copied onto multiple cluster nodes, which ensures that the data is not lost if one node fails.

# YARN Infrastructure: -

- YARN is abbreviated as Yet Another Resource Negotiator.
- Apache Yarn is a part or outside of Hadoop that can act as a standalone resource manager.
- YARN is the framework responsible for providing the computational resources needed for application executions.
- Yarn consists of two important elements are: Resource Manager and Node Manager.

## Resource Manager: -

- One resource manager can be assigned to one cluster per the master.
- Resource manager has the information where the slaves are located and how many resources they have.
- The most important services is the Resource Scheduler that decides how to assign the resources.
- The Resource Manager does this with the Scheduler and Applications Manager

## Node Manager: -

- More than one Node Managers can be assigned to one Cluster.
- Node Manager is the slave of the infrastructure.
- Node Manager takes instructions from the Yarn scheduler to decide which node should run which task.
- The Node Manager reports CPU, memory, disk and network usage to the Resource Manager to decide where to direct new tasks.

# HDFS Federation: -

- HDFS federation is the framework responsible for providing permanent, reliable and distributed storage.
- This is typically used for storing inputs and output (but not intermediate ones).
- It enables support for multiple namespaces in the cluster to improve scalability and isolation.
- In order to scale the name service horizontally, federation uses multiple independent name nodes/namespaces.
- The name nodes are federated, i.e, the name nodes are independent and don’t require coordination with each other.
- The data nodes are used as common storage for blocks by all the name nodes.
- Each data node registers with all the name nodes in the cluster.
- Data nodes send periodic heartbeats and handles commands from the name nodes.

## Storage Solutions: -

- Hadoop uses multiple storage solutions to store and process the data and the techniques

## MapReduce Framework: -

- MapReduce framework is the software layer implementing the MapReduce paradigm.
- Processing can occur on data stored either in a filesystem (unstructured) or in a database (structured).
- A MapReduce framework is usually composed of three steps -

1. **Map:** Each node applies the map function to the local data and writes the output to a temporary storage. A master node ensures that only one copy of redundant input data is processed.

2. **Shuffle:** Each node redistribute data based on the output keys, such that all data belonging to one key is located on the same node.

3. **Reduce:** Each node processes each group of output data, per key, in parallel.

The YARN infrastructure and the HDFS federation are completely decoupled and independent.

The YARN provides resources for running an application while the HDFS federation provides storage.

The MapReduce framework is only one which runs on top of YARN.

# Hadoop - Eco-systems

Hadoop Ecosystem is neither a programming language nor a service.

Hadoop Ecosystem is a platform or framework which solves big data problems.

Hadoop is best known for map reduces and its distributed file system

# HDFS: -

- HDFS abbreviated as Hadoop distributed file system and is the core component of Hadoop Ecosystem.
- HDFS is the primary storage system of Hadoop and distributes the data from across systems.
- HDFS provides scalable, fault tolerance, reliable and cost-efficient data storage for Big data.
- HDFS makes it possible to store several types of large data sets (i.e. structured, unstructured and semi structured data).
- HDFS is a distributed file system that runs on commodity hardware.

- HDFS helps in storing our data across various nodes and maintaining the log file about the stored data (metadata).
- HDFS by default configured for many installations.
- Hadoop interact directly with HDFS by shell-like commands.
- HDFS has two core components, i.e. Name Node and Data Node.
- In HDFS, Name Node stores metadata and Data Node stores the actual data.

# MapReduce: -

- MapReduce is the programming model for Hadoop.
- MapReduce is the core component of processing in a Hadoop Ecosystem as it provides the logic of processing.
- MapReduce is a software framework that helps in writing applications to processes large data sets.
- MapReduce programs runs parallel algorithms in the distributed Hadoop environment.
- MapReduce improves the speed and reliability of cluster using parallel processing.

- MapReduce component has two phases:

  1. Map phase
  2. Reduce phase.

- Each phase has key-value pairs as input and output.

- In addition to the built-in, programmer can also specify two functions:

  1. map function -Map function takes a set of data and converts it into tuples (key/value pairs).

  2. reduce function.- Reduce function takes the output from the Map as an input and combines those data tuples based on the key and accordingly modifies the value of the key.

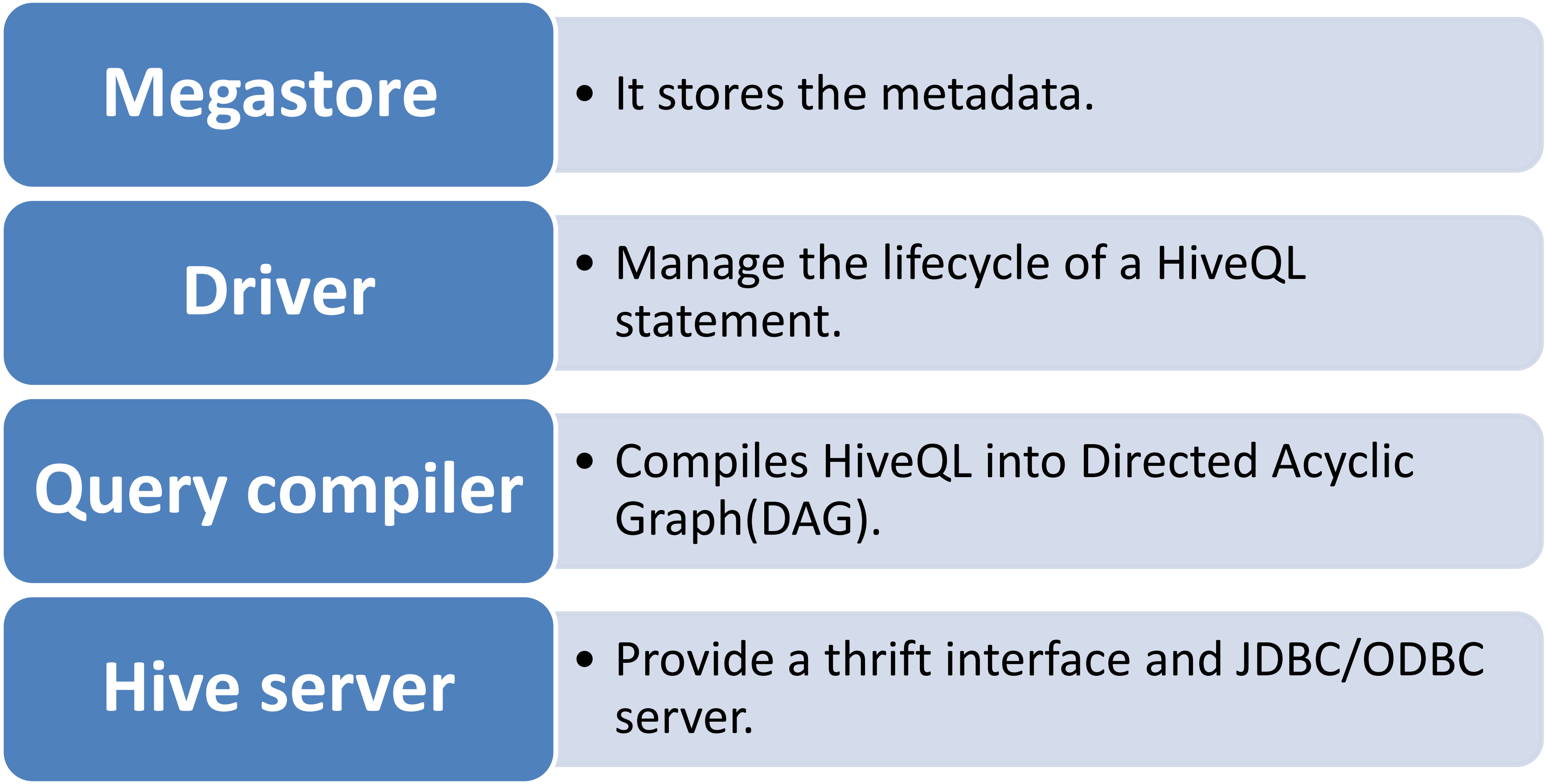
# Hadoop streaming: -

- Hadoop Streaming utility used by developer when they are unable to code map reduce code in other languages.
- Hadoop Streaming is a generic API that allows writing Mappers and Reduces in any language like c, Perl, python, c++ etc.
- Mappers and Reducers receive their input and output on stdin and stdout as (key, value) pairs.
- Streaming is the best fit for text processing.

# Hive: -

- Apache Hive is an open source system for querying and analyzing large datasets stored in Hadoop files.
- HIVE performs reading, writing and managing large data sets in a distributed environment using SQL-like interface.
- Hive use language called Hive Query Language (HQL) that is similar to SQL.
- HiveQL automatically translates SQL-like queries into MapReduce jobs that execute on Hadoop.
- The Hive Command line interface is used to execute HQL commands.
- Hive is highly scalable because of large data set processing and real time processing.
- HiveQL supports all primitive data types of SQL.

Components of hive



**Megastore**

- It stores the metadata.

**Driver**

- Manage the lifecycle of a HiveQL statement.

**Query compiler**

- Compiles HiveQL into Directed Acyclic Graph(DAG).

**Hive server**

- Provide a thrift interface and JDBC/ODBC server.

**Thrift** is an **interface** definition language and binary communication protocol used for defining and creating services for numerous languages.

# Pig: -

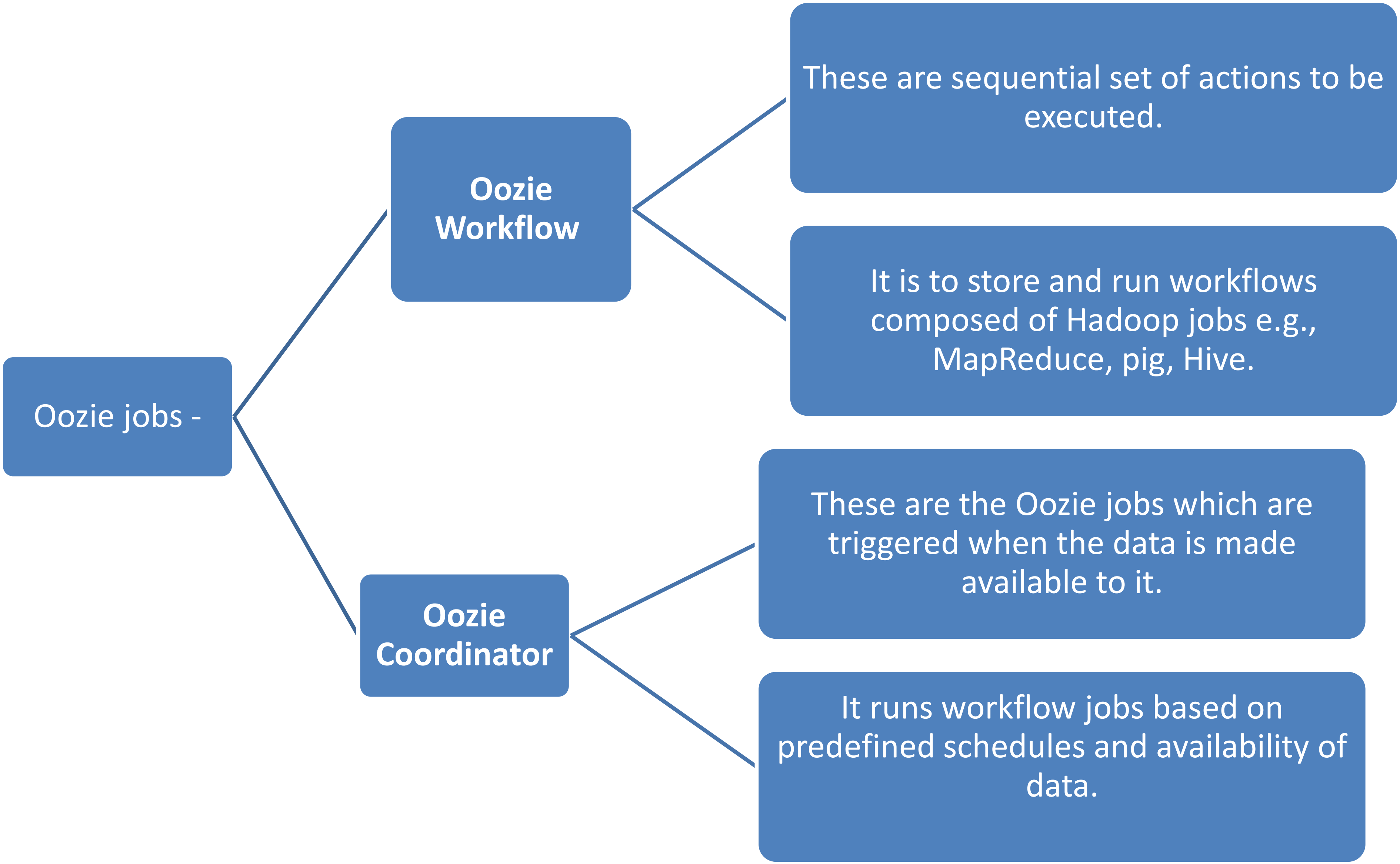
- Apache Pig is a high-level language platform for analyzing and querying large dataset stored in HDFS.
- Pig has two parts: Pig Latin and Pig Runtime.
- Pig Latin is the language and pig runtime is the execution environment.
- Pig Latin language is very similar to SQL.
- It loads the data, applies the required filters and dumps the data in the required format.
- Pig requires Java runtime environment for programs execution.
- Apache Pig features are Extensibility, Optimization opportunities and Handles all kinds of data.
- Pig has incredible price performance and high availability.

# Sqoop: -

- Sqoop imports data from external sources into related Hadoop ecosystem components like HDFS, HBase or Hive.
- Sqoop also exports data from Hadoop to other external sources.
- Map Task is the sub task that imports part of data to the Hadoop Ecosystem.
- When the Job submitted, it is mapped into Map Tasks that brings the chunk of data from HDFS.
- These chunks are exported to the structured data destination.
- Combining all those data chunks, the whole data received at destination.
- Sqoop works with relational databases such as oracle, MySQL.
- Sqoop provides bi-directional data transfer between Hadoop and relational data base.

# Oozie: -

- Oozie is a workflow scheduler system for managing apache Hadoop jobs.
- Oozie combines multiple jobs sequentially into one logical unit of work (UOW).
- For Apache jobs, Oozie has been just like a scheduler.
- Oozie framework is fully integrated with apache Hadoop stack, YARN.
- supports Hadoop jobs for apache MapReduce, Pig, Hive, and Sqoop.
- Oozie is scalable and can manage timely execution of workflows in a Hadoop cluster.
- Oozie is very much flexible because one can easily start, stop, suspend and rerun jobs.
- Oozie provide if-then-else branching and control within Hadoop jobs.



Oozie jobs -

**Oozie Workflow**

These are sequential set of actions to be executed.

It is to store and run workflows composed of Hadoop jobs e.g., MapReduce, pig, Hive.

**Oozie Coordinator**

These are the Oozie jobs which are triggered when the data is made available to it.

It runs workflow jobs based on predefined schedules and availability of data.

# HBase: -

- HBase is an open source, scalable, distributed and non-relational distributed database, i.e. NoSQL database built on top of HDFS.
- HBase was designed to store structured data in tables that could have billions of rows and millions of columns.
- HBase supports all types data including structured, non-structured and semi-structured.
- HBase provides real time access to read or write data in HDFS.
- The HBase was designed to run on top of HDFS to provide Bigtable like capabilities.
- It is accessible through a Java API and has ODBC and JDBC drivers.

- There are two HBase Components namely –
    1. HBase Master
    2. Region Server.

- HBase Master is not part of the actual data storage but negotiates load balancing across all Region Server.

- Region Server is the worker node that handle read, write, update and delete requests from clients.

# Flume: -

- Flume efficiently collecting, aggregating and moving large amount of data from its origin and sending it back to HDFS.
- Flume is distributed, reliable and available service and fault tolerant, reliable mechanism.
- Flume allows the data flow from the source into Hadoop environment.
- Moving data from multiple servers can be done immediately into Hadoop by using Flume.
- Flume also helps to transfer online streaming data from various sources like network traffic, social media, email messages, log files etc. in HDFS.
- Flume is a real time loader for streaming data in to Hadoop.

# Mahout: -

- Mahout is renowned for machine learning.
- Mahout is open source framework for creating scalable machine learning algorithm and data mining library.
- Machine learning algorithms allows to build self-learning machines that evolve by itself without being explicitly programmed.
- Mahout performs collaborative filtering, clustering and classification.
- Mahout provides the data science tools to automatically find meaningful patterns in data stored in HDFS big data sets.
- Mahout used for predictive analytics and other advanced analysis.

Mahout algorithms are –

- **Classification:** It learns from existing categorization and assigns unclassified items to the best category.
- **Clustering:** It takes the item in particular class and organizes them into naturally occurring groups.
- **Collaborative filtering:** It mines user behavior and makes product recommendations.
- **Frequent itemset missing:** It analyzes which objects are likely to be appearing together.
-

# Yarn: -

- YARN is abbreviated as Yet Another Resource Negotiator.
- Apache Yarn is a part or outside of Hadoop that can act as a standalone resource manager.
- YARN is the framework responsible for providing the computational resources needed for application executions.
- Yarn consists of two important elements are: Resource Manager and Node Manager.

## Resource Manager: -

- One resource manager can be assigned to one cluster per the master.
- Resource manager has the information where the slaves are located and how many resources they have.
- Resource manager runs several services.
- The most important services is the Resource Scheduler that decides how to assign the resources.
- The Resource Manager does this with the Scheduler and Applications Manager.

## Node Manager: -

- More than one Node Managers can be assigned to one Cluster.
- Node Manager is the slave of the infrastructure.
- Node Manager sends a heartbeat to the Resource Manager periodically.
- Node Manager takes instructions from the Yarn scheduler to decide which node should run which task.
- The Node Manager reports CPU, memory, disk and network usage to the Resource Manager to decide where to direct new tasks.

# HCatalog: -

- HCatalog is a Hadoop storage and table management layer.
- HCatalog enables different data processing tools like Pig, MapReduce for Users.
- Users can easily read and write data on the grid by using the tools enabled by HCatalog.
- Users can directly load the tables using pig or MapReduce and no need to worry about re-defining the input schemas.
- HCatalog exposes the tabular data of HCatalog meta store to other Hadoop applications.
- Apache HCatalog is a project enabling non-HCatalog scripts to access HCatalog tables.

- The users need not worry about where or in what format their data is stored.

- HCatalog table concept provides a relational view of data in the Hadoop Distributed File System (HDFS) to the users.

- HCatalog can displays data from RCFile format, text files, or sequence files in a tabular view.

- HCatalog also provides APIs to access these tables metadata by external systems.

# Spark: -

- Apache Spark is both a programming model and a computing model framework for real time data analytics in a distributed computing environment.
- It executes in-memory computations to increase speed of data processing over Map-Reduce which is a big reason for its popularity.
- Spark is an alternative to MapReduce that enables workloads to execute in memory instead of on disk.
- By using in-memory computing, Spark workloads typically run between 10 and 100 times faster compared to disk execution.
- Spark can be used independently of Hadoop.
- Spark supports SQL that helps to overcome a short coming in core Hadoop technology.
- The Spark programming environment works with Scala, Python and R shells interactively.

# Apache Drill: -

- Apache Drill processes large-scale data including structured and semi-structured data.
- Apache Drill is low latency distributed query engine designed to scale several thousands of nodes and query petabytes of data.
- The drill has specialized memory management system to eliminates garbage collection and optimize memory allocation and usage.
- Apache Drill is used to drill into any kind of data.

- Drill is an open source application works well with Hive by allowing developers to reuse their existing Hive deployment.

- The main power of Apache Drill lies in combining a variety of data stores just by using a single query.

- Apache Drill features are Extensibility, flexibility, drill decentralized metadata and dynamic schema discovery.
-

# Ambari: -

- Ambari is a management platform for provisioning, managing, monitoring and securing apache Hadoop cluster.
- This includes support for Hadoop HDFS, Hadoop MapReduce, Hive, HCatalog, HBase, Zookeeper, Oozie, Pig, and Sqoop.
- Ambari provide consistent, secure platform for operational control.
- Ambari features are Simplified installation, configuration and management, Centralized security setup, Highly extensible and customizable and Full visibility into cluster health.

# Zookeeper: -

- Apache Zookeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization and group services.
- Zookeeper manages and coordinates with various services in a distributed environment.
- It saves a lot of time by performing synchronization, configuration maintenance, grouping and naming.
- Zookeeper is fast with workloads where reading data are more common than writing data.
- Zookeeper maintains a record of all transactions.