EGT 305: BIG DATA PROCESSING & APPLICATION

BOSS

TOPIC 1:

INTRODUCTION TO BIG DATA AND IT'S APPLICATION

- DEFINITION OF BIG DATA
- WHAT MAKES BIG DATA
- BIG DATA APPLICATIONS AND PRACTICES
- OVERVIEW OF BIG DATA FRAMEWORK

DATA SIZE

Term	Abb	Size
Kilobyte	KB	1000 Bytes
Megabyte	MB	1000 KB
Gigabyte	GB	1000 MB
Terabyte	TB	1000 GB
Petabyte	PB	1000 TB
Exabyte	EB	1000 PB
Zettabyte	ZB	1000 EB
Yottabyte	YB	1000 ZB

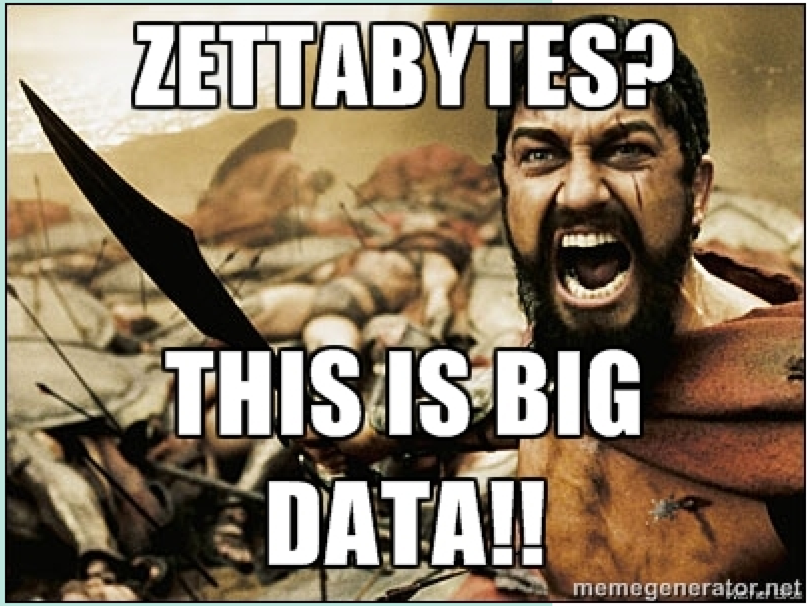
Apollo Guidance Computer had RAM of 4KB, a 32KB hard disk.

Falcon 9 rocket with terabytes of RAM and hard disk.

WHAT IS BIG DATA

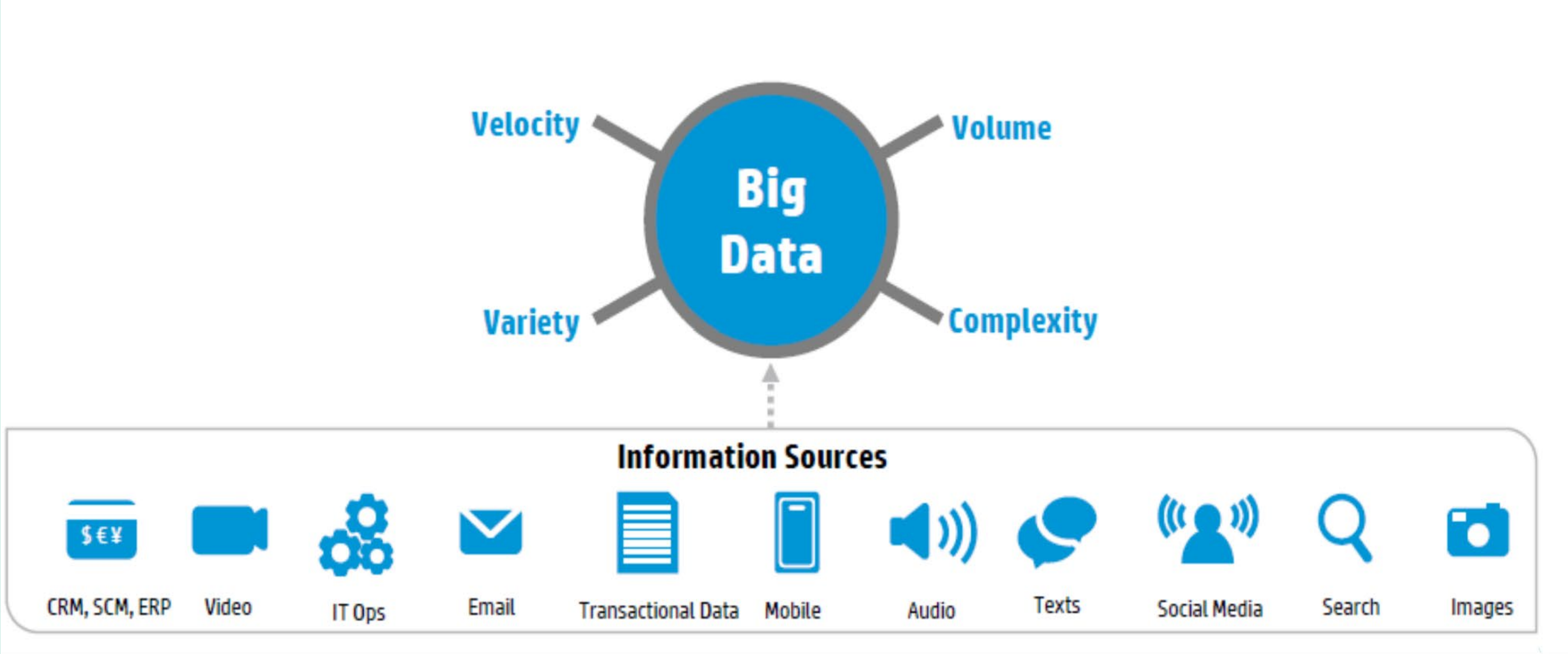
Define: *Big data is data that contains greater **variety**, arriving in increasing **volumes** and with more **velocity**. This is also known as the three Vs. (From Oracle)*

Put simply, big data is larger, more complex data sets, especially from new data sources that provides a lot of **variety.** These data sets are so **voluminous** that traditional data processing software just can't manage them. But these massive **volumes** of data can be used to address business problems you wouldn't have been able to tackle before. These data are collected at rapid speed due to ease of collection.



Source: https://www.oracle.com/sg/big-data/what-is-big-data/#:~:text=What%20exactly%20is%20big%20data,especially%20from%20new%20data%20sources

WHY IS IT CALLED BIG DATA?



NEW INNOVATIONS
iCatbot
Save time & energy cleaning your home with the ultimate pet-powered cleaning robot*
iCatbot
Cat not included
LEARN MORE ▸
$199.99
#412014 | iCatbot Pet-Powered Cleaning Robot

Information Sources: It can be anything that collects data from your TikTok search history to your vacuum cleaner

Source: Big Data Use Cases from HP

WHY IS IT IMPORTANT?

How you use it makes it important.

Analyzing it is important. Allowing us to

- streamline resource management
- improve operational efficiencies
- optimize product development
- drive new revenue and growth opportunities
- enable smart decision making.

Source:
https://www.sas.com/en_sg/insights/big-data/what-is-big-data.html

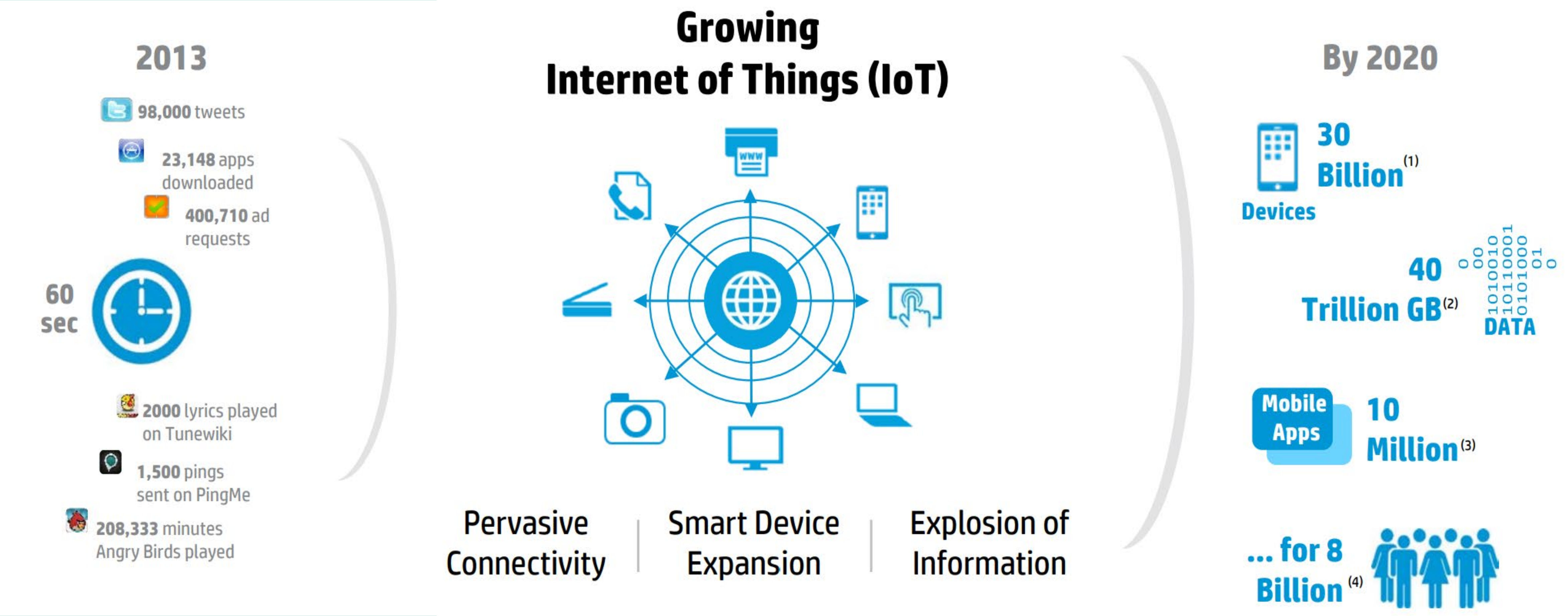
WHY IS IT IMPORTANT?

Combining it with high-performance AI models, you can accomplish tasks such as:

- Determining root causes of failures, issues and defects in near-real time.
- Spotting anomalies faster and more accurately than the human eye.
- Improving patient outcomes by rapidly converting medical image data into insights.
- Recalculating entire risk portfolios in minutes.
- Improving deep learning models' ability to accurately classify and react to changing variables.
- Detecting fraudulent behavior before it affects your organization..

Source:
https://www.sas.com/en_sg/insights/big-data/what-is-big-data.html

DATA CAN BE COLLECTED ANYWHERE NOW



2013

98,000 tweets

23,148 apps downloaded

400,710 ad requests

60 sec

2000 lyrics played on Tunewiki

1,500 pings sent on PingMe

208,333 minutes Angry Birds played

Growing
Internet of Things (IoT)

Pervasive Connectivity | Smart Device Expansion | Explosion of Information

By 2020

30 Billion (1) Devices

40 Trillion GB (2) DATA

Mobile Apps 10 Million (3)

... for 8 Billion (4)

Source: Big Data Use Cases from HP

SOURCES OF DATA AND APPLICATIONS

Industry	Data Produced	Applications
Aerospace	Movement of satellites and stars or engine and flight parameters data from planes	To monitor movement of asteroid activities to engine predictive maintenance
Financial institute	News content via video, audio, twitter and news report	To make trading decisions
Healthcare	Medical records and images	To aid in short-term public health monitoring and long-term epidemiological research
IoTs	Sensor data	Monitor activities in smart cities
Media/Entertainment	Content and user viewing behavior	To capture more viewers by creating content based on the audience preference
Social Media	TikTok, IG posts, social networking sites, log details	To analyze the customer behavior pattern
Transportation, Logistics, Retail, Utilities	Sensor data generated from fleet transceivers, RFID tag readers and smart meters	To optimize operations

HOW TO STORE LARGE DATASETS?

First thing, how to **store** them?

Single machine? 60TB SSD announced. $$$$$...

Cluster of machines?

- How many machines?
- Need data backup, redundancy, recovery, etc.
- Need to worry about machine and drive failure.

Source: Big Data Use Cases from HP

HOW OFTEN TO DISK FAIL?

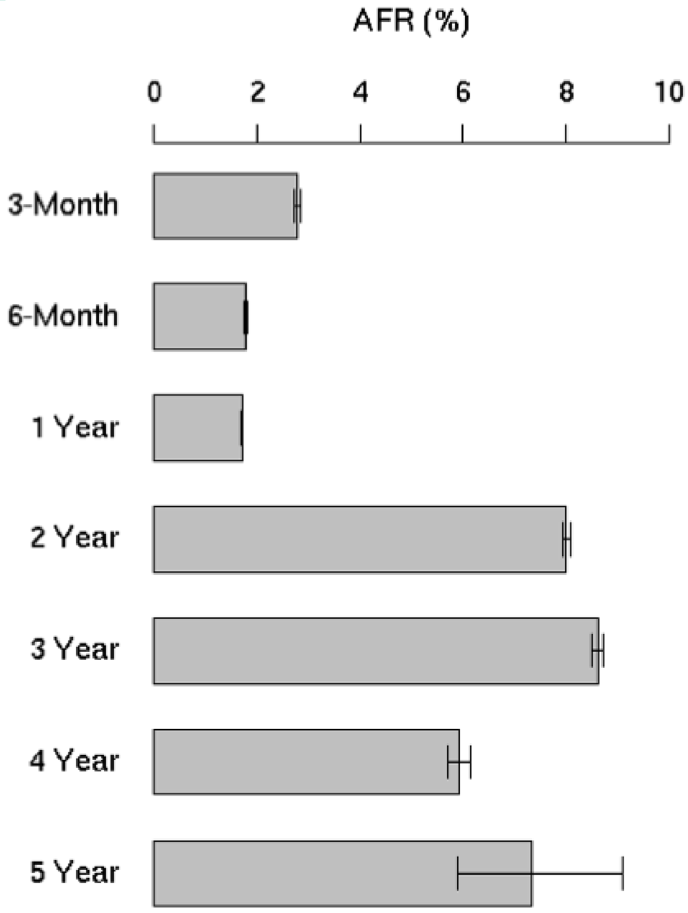


Figure 2: Annualized failure rates broken down by age groups

3% of 100,000 hard drives fail within first 3 months

Source: Big Data Use Cases from HP

CASE STUDY: TRADITIONAL SOLUTION VS BIG DATA SOLUTION

Why is Big Data Solution more advantageous compared to normal traditional solution of processing (What you have learnt in year 2).

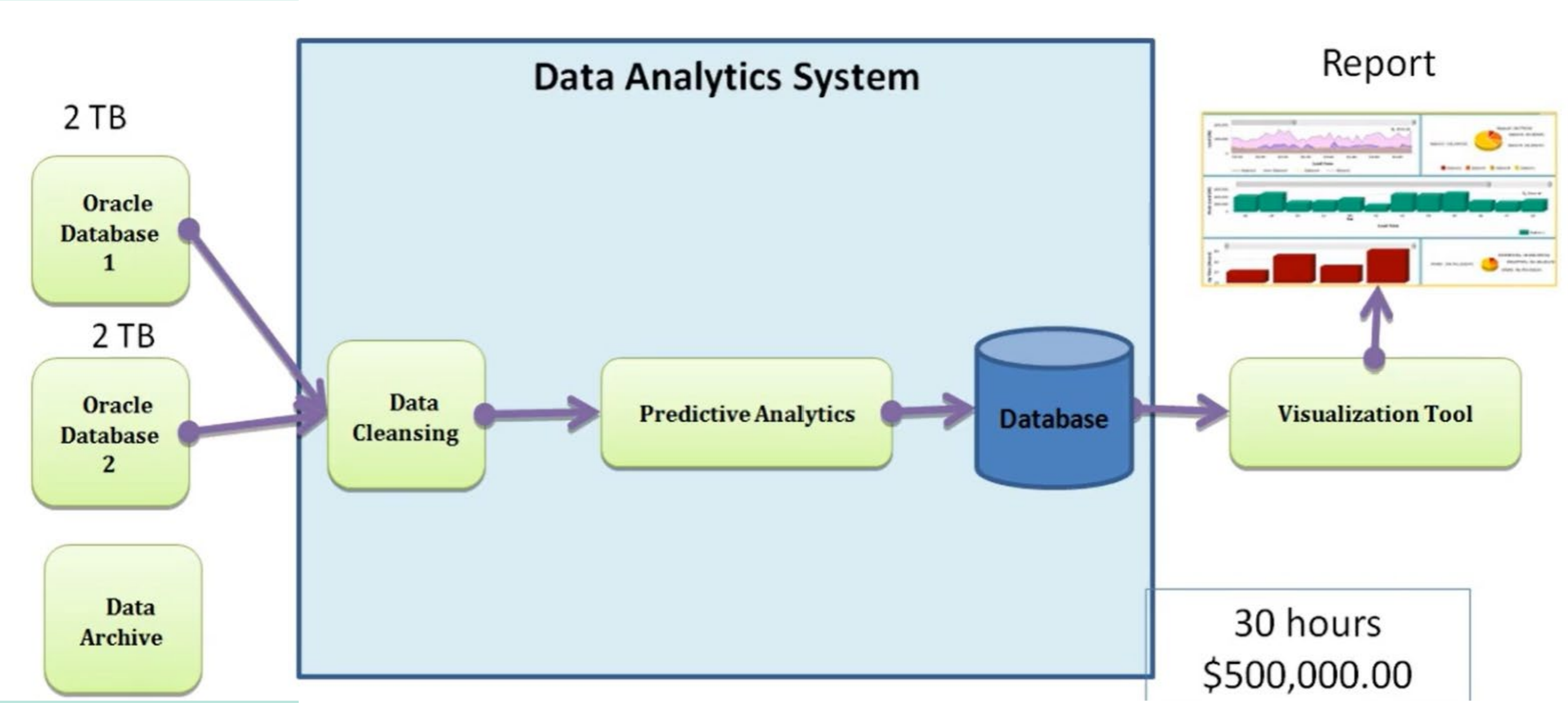
Context: Analyze late payment risk

A fast-food chain food supplier wants to match invoice to payments received from vendors to predict payment risks.

The invoice system and receivable system are maintained in Oracle (Enterprise Resource Planning) software. They want to present the data using PowerBI, which is a data visualization tool.

CASE STUDY: TRADITIONAL SOLUTION VS BIG DATA SOLUTION

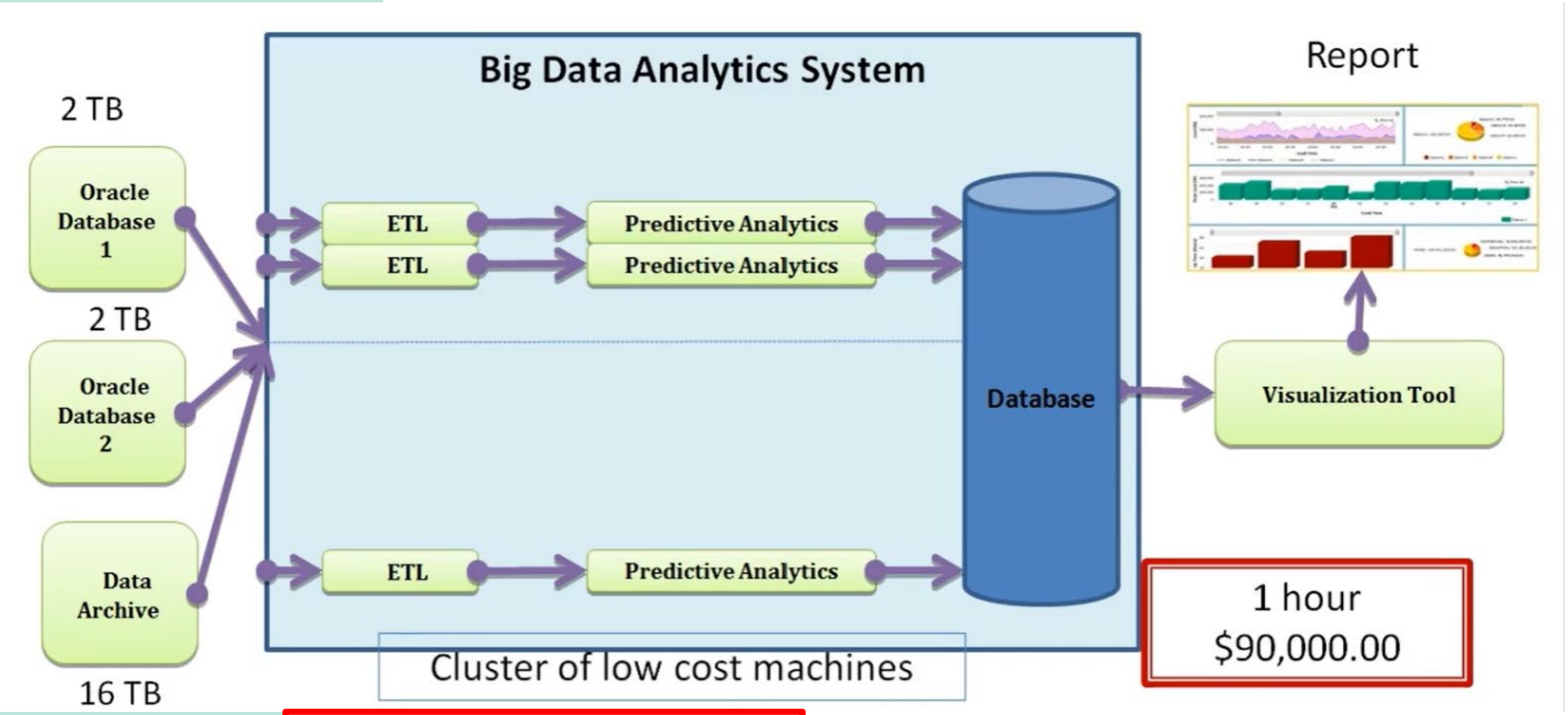
Traditional Solution



20TB (NOT USED)

CASE STUDY: TRADITIONAL SOLUTION VS BIG DATA SOLUTION

Big Data Solution



Big Data Analytics System

2 TB

Oracle Database 1

2 TB

Oracle Database 2

Data Archive

16 TB

ETL

ETL

ETL

Predictive Analytics

Predictive Analytics

Predictive Analytics

Database

Report

Visualization Tool

Cluster of low cost machines

1 hour
$90,000.00

*Extract, transform, and load (ETL)

CASE STUDY: TRADITIONAL SOLUTION VS BIG DATA SOLUTION

	Traditional Solution	**Big Data Solution**
Data Source	Two Oracle databases, each 2 Terabytes (4 TB)	Two Oracle databases (2 Terabytes each) and 16 Terabytes of archived data(20TB)
Infrastructure	Single cluster of high end computing server	Cluster of 30 low-cost commodity class machines.
Data Cleansing	ETL process is done line by line and column by column.	Parallel ETL processing is done (Data is processed using open-source big data tools.)
Output	Predictions stored in MySQL database	Processed data stored in MySQL database
Cost	Upwards of $500,000	Reduced to about $90,000 (80% cost reduction)
Performance	Takes 30 hours to generate a report	Processing time reduced to little over an hour (30 times faster)
Open Source Tools	Relies on costly software	Relies on free and low-maintenance open-source big data such as Hadoop or Spark.
Fault Tolerance	Not explicitly mentioned	Built-in fault tolerance and data replication

BIG DATA FRAMEWORK

HOW TO ANALYZE LARGE DATASETS?

WHAT SOFTWARE LIBRARIES TO USE?

ANOTHER PROGRAMMING LANGUAGE TO LEARN? (R, JAVA?)

IS THERE A GUIDE?

BIG DATA FRAMEWORK

FRAMEWORK

- Frameworks simplify big data processing for rapid, effective, and secure analysis.
- Often open-source, providing cost-effectiveness with the option to purchase support if needed.
- Examples of framework are:

APACHE Spark™

hadoop

APACHE STORM™

HIVE

APACHE HBASE

PURPOSE OR REASONS

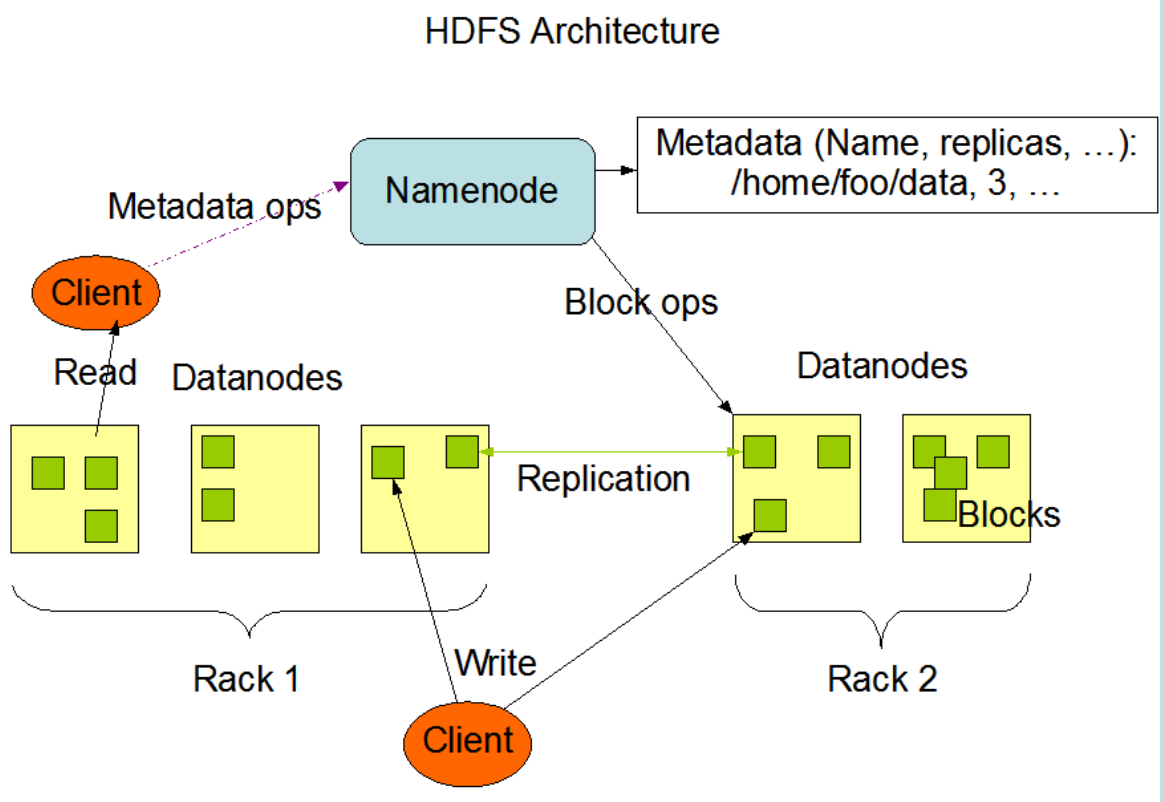
- Aim to offer organization and structure for corporate companies leveraging the potential of big data.
- Designed to address the challenges of integrating successful big data practices into businesses.
- Acknowledges the need for structure, skills, talented personnel, and cutting-edge technology for long-term success.
- Provides a strategy encompassing all organizational capabilities necessary for a fruitful big data practice.
- Addresses everything from formulating a big data strategy to acquiring the required technical equipment and skills.

APACHE HADOOP

- Open-sourced software (Low cost)
- For distributed computing
  - Breaks down the data into small chunks
  - Distribute these chunks across many machines
  - Processed by this machine.
- Scalable:
  - Can be scaled to thousands of machines
  - Linear scalability or parallel running. Example if you have two machine, your job run 2 as fast
- File system called Hadoop Distributed File System (HDFS):
  - Able to recover from machine/disk failure (Do not need to restart computation)
  - Failure resistant: This fault resistance is provided by replicating the data to another node in the cluster, thus in the event of a failure, the data from the replicated node can be used, thereby maintaining data consistency.
- Uses a simple programming model, MapReduce:
  - MapReduce in Hadoop is a distributed programming model that processes large datasets by dividing them into smaller chunks, applying a user-defined map function, shuffling and sorting the intermediate results, and then applying a reduce function to generate the final output.
- Written in Java

HDFS ARCHITECTURE

Divides files into blocks, replicates them across nodes for fault tolerance, and enables parallel processing, ensuring scalability and reliability for big data storage and retrieval.



What is
Hadoop?
1990
January
IBM
simplilearn

APACHE SPARK

- Open-sourced software (Low cost)
- Scalable
- A lightning-fast cluster computing technology, designed for fast computation
- In-memory cluster computing that increases the processing speed of an application
  - Data stored in RAM allowing for faster data access compared to in disk storage
  - It is up to 100 times faster in memory, and 10 times faster when running on disk.
  - Possible by reducing number of read/write operations to disk. It stores the intermediate processing data in memory.
- Real-time analytics
  - Speed of in-memory computing makes it well-suited for real-time analytics and processing of large datasets, enabling rapid insights and decision-making.
- NOT a modified version of Hadoop
- Spark uses Hadoop in two ways – one is storage and second is processing
  - Hadoop MapReduce and it extends the MapReduce model to efficiently use it for more types of computations, which includes interactive queries and stream processing.
- Supports multiple languages: APIs in Java, Scala, or Python.
- Advanced Analytics **–** Spark not only supports 'Map' and 'reduce'. It also supports SQL queries, Streaming data, Machine learning (ML), and Graph algorithms.

APACHE SPARK VS HADOOP

1.Purpose and Functionality:

- Hadoop: Primarily designed for **batch processing**.
- Spark: Versatile, supporting batch processing, **real-time stream processing**, and **interactive queries**.

2.Data Processing Paradigm:

- Hadoop: Processes data in **batches**, reading from disk.
- Spark: Processes data **in-memory**, speeding up computations.

3.Fault Tolerance:

- Hadoop: Ensures fault tolerance by creating **multiple copies** of data blocks.
- Spark: Uses **Resilient Distributed Dataset (RDD)** for fault tolerance.

4.Resource Utilization:

- Hadoop: Processes data using **ordinary machines**, less resource-intensive.
- Spark: Leverages **in-memory caching**, requires significant **RAM**.

5.Advanced Features:

- Hadoop: Robust but lacks advanced features.
- Spark: More advanced, incorporates **AI/ML** capabilities.

6.Integration:

- Many organizations use **both Spark and Hadoop** together.

What is
Apache
Spark?
IBM Cloud
Apache Spark

APACHE HIVE

- Purpose: Apache Hive is a data warehousing and SQL-like query language system built on top of Hadoop for managing and querying large datasets.
- Data Storage: Utilizes Hadoop Distributed File System (HDFS) for distributed and scalable storage of structured and semi-structured data.
- HiveQL: Allows users to query data using a SQL-like language called Hive Query Language (HiveQL), making it more accessible for those familiar with SQL.
- Metadata Storage: Maintains metadata about tables, partitions, and columns in a metastore, allowing for efficient query planning and optimization.
- Integration: Integrates with other Hadoop ecosystem tools and frameworks, such as Apache HBase, Apache Spark, and Apache HCatalog.
- User-Defined Functions (UDFs): Supports the use of User-Defined Functions to extend functionality and enable custom processing logic.

APACHE HBASE

NoSQL Database:

- Apache HBase is a distributed, scalable, and open-source NoSQL database that is designed to provide random access to large amounts of structured data, making it suitable for real-time read and write operations.

Built on Hadoop:

- HBase is built on top of the Hadoop Distributed File System (HDFS) and is part of the Apache Hadoop project, leveraging Hadoop's distributed and fault-tolerant architecture.

Columnar Store:

- It organizes data in tables with rows and columns, similar to a relational database, but with a key difference—it uses a columnar store, allowing for efficient retrieval of specific columns of data. (NoSQL)

Scalability and High Availability:

- HBase is designed for horizontal scalability, enabling the addition of more nodes to handle increased data and workload. It also provides high availability by replicating data across multiple nodes, ensuring fault tolerance.

Use Cases:

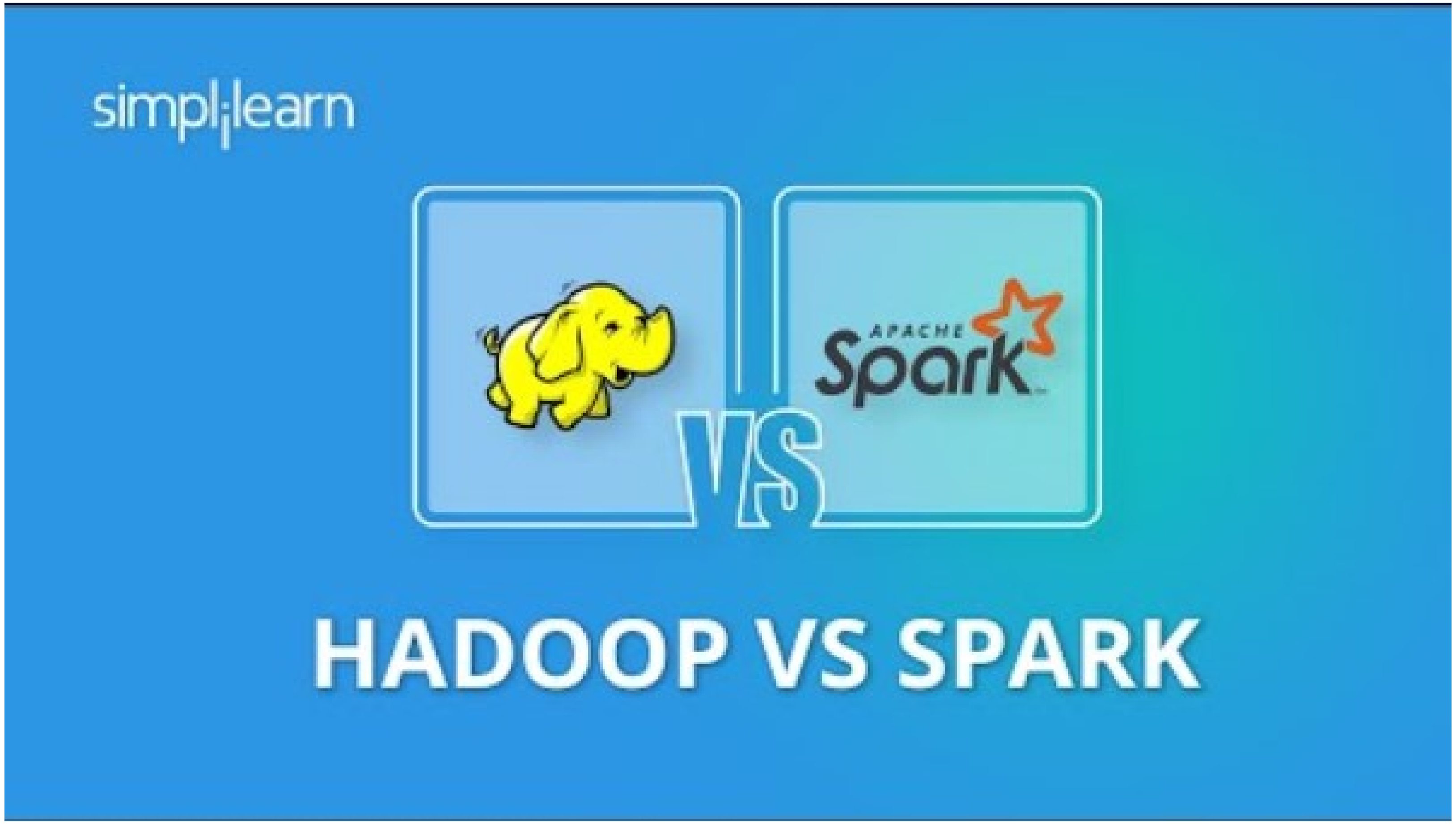
- Apache HBase is well-suited for applications that require real-time, random access to large datasets, such as online transactional processing (OLTP) systems, time-series data, sensor data, and applications requiring low-latency access to data.

IN A NUTSHELL

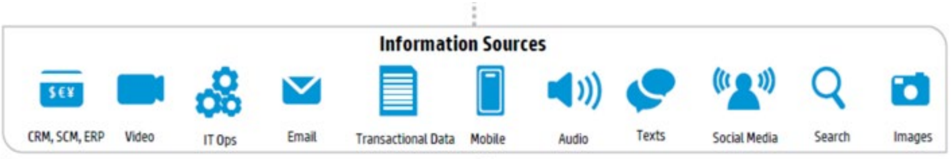
Feature	Hadoop	Spark	Hive	HBase
Type	Framework for distributed storage and processing	In-memory cluster computing framework	Data warehousing and SQL-like query system	Distributed, scalable NoSQL database
Primary Purpose	Distributed storage and batch processing	Fast in-memory data processing	SQL-like querying on large datasets	Real-time, random access to large datasets
Processing Model	Batch processing using MapReduce	Batch and real-time processing	Batch processing using HiveQL queries	Real-time read and write operations
Data Processing	MapReduce	RDDs (Resilient Distributed Datasets)	Translates HiveQL into MapReduce or Tez tasks	Columnar store with key-value pair storage
Query Language	N/A (MapReduce programming model)	Spark SQL and DataFrame API	HiveQL (SQL-like language)	N/A (Programmatic API and client libraries)
Data Storage	Hadoop Distributed File System (HDFS)	Various storage options including HDFS	HDFS, or other file systems	HDFS, and HBase native storage
Use Cases	Batch processing, large-scale data storage	Real-time data processing, iterative algorithms	Data warehousing, SQL-like querying	Real-time applications, random data access
Programming Model	Java-based MapReduce programming model	Supports multiple languages (Scala, Java, Python, etc.)	HiveQL (SQL-like queries)	Java API, RESTful API, Thrift API

rapidminer
Apache Solr
IBM Watson
Dato
elastic
clarifai
Lucene
Spark
ORACLE NOSQL DATABASE
cassandra
APACHE HBASE
TEZ
CouchDB
mongoDB
NumPy
MADlib
APACHE STORM™
HIVE
MapReduce
SciPy
hadoop
DATASTAX
influxdata
MATLAB
SciDB
MESOS
TACHYON
sas
OOZIE
APACHE DRILL
Apache Ambari
nifi
Couchbase
R
SPSS
DATATORRENT
redislabs
amazon DynamoDB
accumulo
mllib
plotly
Scala
kibana
FLUME
kognitio
HPCC SYSTEMS®
mahout
data iku
TensorFlow
splunk>
sumologic
MariaDB
TERADATA
NUODB
VERTICA
WEKA The University of Waikato
sense
OrientDB®
INFINITEGRAPH®
Pivotal.
Clustrix
dremio
NLTK
CONTINUUM
DataRobot
neo4j
APACHE GIRAPH
GraphLab
memsql
Greenplum
NETEZZA
SAP HANA
Data Flair

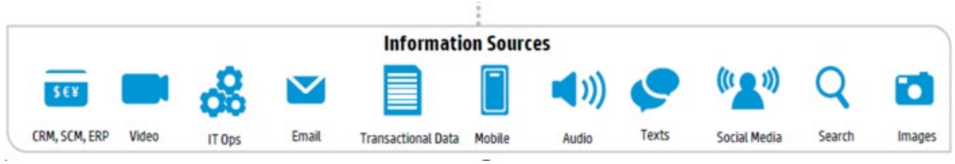
ADDITIONAL RESOURCES FOR INSOMNIA



MORE INFO ON BIG DATA FRAMEWORKS



MORE INFO ON BIG DATA FRAMEWORKS



APACHE Spark™ hadoop

Frameworks allow you to easily do the following:

- Store (LEGO store it nicely into boxes)
- Process (Building the stuff)
- Analyze (Determining which is the most idea way to build)
- Visualize (Showing off your creation)

MORE INFO ON BIG DATA FRAMEWORKS (SIMPLIFIED)

hadoop

- Sorter for data. Which allows distributed processing(**Not processed in a single machine**) of massive data sets across clusters of computers.

Key Components:

- **HDFS**: Imagine a giant storage room where you can stash petabytes of data. (Store room)
- **MapReduce:** Divides the data into smaller chunks, process it, shuffling and sorting the intermediate results, and a reduce function to generate the final output (Worker who hard carries)
- **YARN:** Manager that handles resources. (Boss who delegates the work)

APACHE Spark

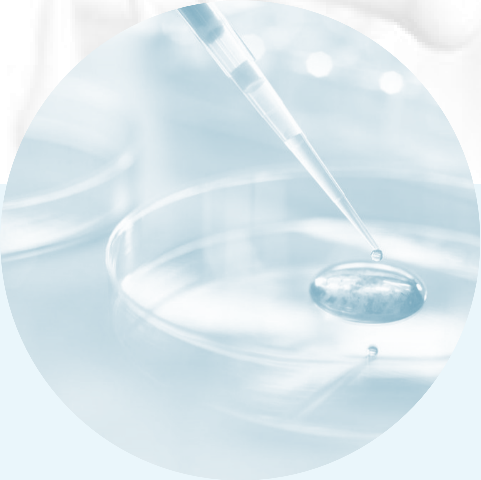
- Versatile engine for data processes. It can do data engineering, science, and machine learning on single machines or clusters.

Features:

- **Batch/Streaming Data**: Batch processing (like sorting LEGO bricks) and real-time streaming (like building a LEGO castle).
- **SQL Analytics**: SQL queries for dashboards and reports.
- **Data Science at Scale**: Imagine exploring petabyte-scale data without downsampling. (Reducing the data size)
- **Machine Learning**: Train algorithms
- **Languages Supported:** Python, SQL, Scala, Java, and R

NOTE

- We would not be installing or running any of this on your notebook
- Mostly working on Spark
- If you would like to try out and install Spark,Hive,Hadoop or Hbase on your notebook please find online resources for the step-by-step guide.



THANK YOU

SLEEPYHEAD
SLEPPYMEL@GMAIL.COM