High Performance and Distributed Computing for Big Data

Unit 3: Big Data - AWS EMR

Jordi Mateo Fornés jordi.mateo@udl.cat

Universitat Rovira i Virgili and Universitat de Lleida

Today's lecture

1. Introduction to Big Data
2. Hadoop and MapReduce
3. HandsOn: 1000 Genomes - Population Distribution

Big Data

What is data?

*Data is a set of values of **qualitative** or **quantitative** variables about one or more persons or objects, while a datum (singular of data) is a single value of a single variable. Wikipedia*

What is data?

*Data is a set of values of **qualitative** or **quantitative** variables about one or more persons or objects, while a datum (singular of data) is a single value of a single variable. Wikipedia*

Patient ID	Name	Height	Weight	Age
1	John	1.80	80	30
2	Mary	1.60	60	25
3	Paul	1.70	70	35
4	Jane	1.65	65	40

- Rows: Objects, Samples, Observations, Individuals
- Columns: Variables, Features, Attributes, Dimensions

Which is bigger?

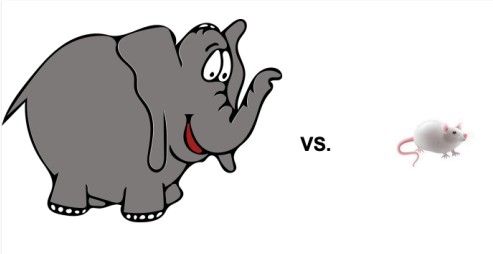


Figure 1: Example extracted from: Introduction to Big Data (Harvard)

Is it bigger an elephant or a mouse?

Which is bigger?

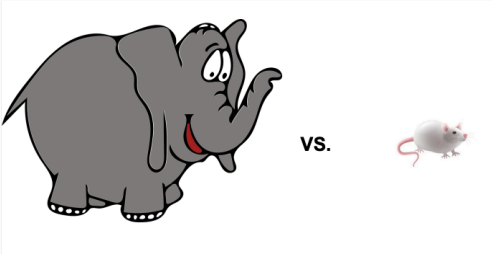


Figure 1: Example extracted from: Introduction to Big Data (Harvard)

Is it bigger an elephant or a mouse?

YES - NO - DEPENDS ⇒ Depends on Complexity, Variety, Velocity, Veracity, Amount

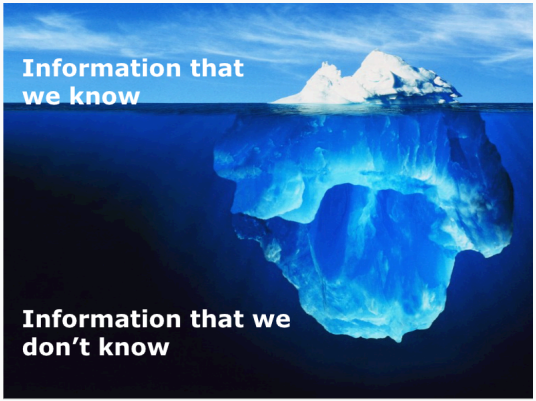
What is Big Data?

Concept

Big Data refers to extremely large, complex, and diverse datasets that cannot be effectively managed, processed, and analyzed using traditional data processing techniques

- Big data means **sample size** and **dimensionality**.
  - **Sample size**: The number of observations in a dataset.
  - **Dimensionality**: The number of variables in a dataset.
- Big data means **complexity** and **variety**.
  - **Complexity**: The number of different types of data in a dataset.
  - **Variety**: The number of different sources of data in a dataset.

Iceberg Analogy



What are the 5 V's of Big Data?

- **Volume**: The amount of data (Data at Rest)⇒ Terabytes to Zettabytes of data generated everday and stored in data centers.

What are the 5 V's of Big Data?

- **Volume**: The amount of data (Data at Rest)⇒ Terabytes to Zettabytes of data generated everday and stored in data centers.
- **Velocity**: The speed at which data is generated and processed (Data in Motion) ⇒ Real-time data processing from seconds to milliseconds response time.

What are the 5 V's of Big Data?

- **Volume**: The amount of data (Data at Rest)⇒ Terabytes to Zettabytes of data generated everday and stored in data centers.
- **Velocity**: The speed at which data is generated and processed (Data in Motion) ⇒ Real-time data processing from seconds to milliseconds response time.
- **Variety**: The different types of data (Data in Many Forms) ⇒ Structured, unstructured, semi-structured data, text, images, audio, video, ...

What are the 5 V's of Big Data?

- **Volume**: The amount of data (Data at Rest)⇒ Terabytes to Zettabytes of data generated everday and stored in data centers.
- **Velocity**: The speed at which data is generated and processed (Data in Motion) ⇒ Real-time data processing from seconds to milliseconds response time.
- **Variety**: The different types of data (Data in Many Forms) ⇒ Structured, unstructured, semi-structured data, text, images, audio, video, ...
- **Veracity**: The quality of the data (Data in Doubt) ⇒ Trustworthiness, accuracy, and reliability of the data.

What are the 5 V's of Big Data?

- **Volume**: The amount of data (Data at Rest)⇒ Terabytes to Zettabytes of data generated everday and stored in data centers.
- **Velocity**: The speed at which data is generated and processed (Data in Motion) ⇒ Real-time data processing from seconds to milliseconds response time.
- **Variety**: The different types of data (Data in Many Forms) ⇒ Structured, unstructured, semi-structured data, text, images, audio, video, ...
- **Veracity**: The quality of the data (Data in Doubt) ⇒ Trustworthiness, accuracy, and reliability of the data.
- **Value**: The insights that can be derived from the data (Data in Action) ⇒ The ability to turn data into value.

Examples of V's of Big Data

Volume

- *Dr. Katie Bouman* and the hard drives used on the Event Horizon telescope (2019). A total of 5 petabytes of data was collected. With this data, the **first-ever image of a black hole** was created.

Examples of V's of Big Data

Volume

- *Dr. Katie Bouman* and the hard drives used on the Event Horizon telescope (2019). A total of 5 petabytes of data was collected. With this data, the **first-ever image of a black hole** was created.

Velocity

- **Elon Musk** makes a tweet on 2022 saying: *World Cup traffic hit almost 20000 tweets per second today!*
- Probably we should take a decision when we have around 70% of the data we would like to have. If we wait until 90%, in many cases, we are probably being too slow.

Examples of V's of Big Data

Volume

- *Dr. Katie Bouman* and the hard drives used on the Event Horizon telescope (2019). A total of 5 petabytes of data was collected. With this data, the **first-ever image of a black hole** was created.

Velocity

- **Elon Musk** makes a tweet on 2022 saying: *World Cup traffic hit almost 20000 tweets per second today!*
- Probably we should take a decision when we have around 70% of the data we would like to have. If we wait until 90%, in many cases, we are probably being too slow.

Variety

- **Healthcare data**: Electronic Health Records (EHR), Medical Imaging, Genomic Data, Respiratory Sounds, ...

Examples of V's of Big Data

Volume

- *Dr. Katie Bouman* and the hard drives used on the Event Horizon telescope (2019). A total of 5 petabytes of data was collected. With this data, the **first-ever image of a black hole** was created.

Veracity

- **December 2009**: *HP* investigates instances of "racist" webcams. The webcams were unable to detect the faces of dark-skinned individuals.
- **May 2016**: *ProPublica* investigation finds that a software used across the US to predict future criminals is biased against black.

Velocity

- **Elon Musk** makes a tweet on 2022 saying: *World Cup traffic hit almost 20000 tweets per second today!*
- Probably we should take a decision when we have around 70% of the data we would like to have. If we wait until 90%, in many cases, we are probably being too slow.

Variety

- **Healthcare data**: Electronic Health Records (EHR), Medical Imaging, Genomic Data, Respiratory Sounds, ...

Bytes and Beyond

Approximately *328.77 million* terabytes of data are created each day.

Units of Data

- **Yottabyte**: 1,000 Zettabytes
- **Zettabyte**: 1,000 Exabytes
- **Exabyte**: 1,000 Petabytes
- **Petabyte**: 1,000 Terabytes
- **Terabyte**: 1,000 Gigabytes
- **Gigabyte**: 1,000 Megabytes

Questions

- How to handle such a large amount of data?
- How to store, process, and analyze it?
- How to extract value from it?
- Is all this data useful?

What are the challenges of Big Data?

- **Data Sampling**: What are the characteristics of my data samples?

What are the challenges of Big Data?

- **Data Sampling**: What are the characteristics of my data samples?
- **Data Sources**: What are the origins of my data?

What are the challenges of Big Data?

- **Data Sampling**: What are the characteristics of my data samples?
- **Data Sources**: What are the origins of my data?
- **Data Capture**: What specific data do we aim to capture?

What are the challenges of Big Data?

- **Data Sampling**: What are the characteristics of my data samples?
- **Data Sources**: What are the origins of my data?
- **Data Capture**: What specific data do we aim to capture?
- **Data Usability**: Is the captured data useful and usable for our purposes?

What are the challenges of Big Data?

- **Data Sampling**: What are the characteristics of my data samples?
- **Data Sources**: What are the origins of my data?
- **Data Capture**: What specific data do we aim to capture?
- **Data Usability**: Is the captured data useful and usable for our purposes?
- **Algorithm Selection**: What are the most effective algorithms for our data?

What are the challenges of Big Data?

- **Data Sampling**: What are the characteristics of my data samples?
- **Data Sources**: What are the origins of my data?
- **Data Capture**: What specific data do we aim to capture?
- **Data Usability**: Is the captured data useful and usable for our purposes?
- **Algorithm Selection**: What are the most effective algorithms for our data?
- **Quantity vs Quality**:
  - Does bigger data necessarily mean better results?
  - What has a greater impact, the quality or the quantity of data?

What are the challenges of Big Data?

- **Data Sampling**: What are the characteristics of my data samples?
- **Data Sources**: What are the origins of my data?
- **Data Capture**: What specific data do we aim to capture?
- **Data Usability**: Is the captured data useful and usable for our purposes?
- **Algorithm Selection**: What are the most effective algorithms for our data?
- **Quantity vs Quality**:
  - Does bigger data necessarily mean better results?
  - What has a greater impact, the quality or the quantity of data?
- **Metrics Selection**: Which metrics should we use to measure performance, efficiency, scalability, and usability?

To recap. Big Data is...

When...

- ... data that will not fit in main memory,
- ... data cannot be handled by traditional data processing techniques
- ... a calculation needs more than a single computer to process.
- ... a dataset is too large to fit into the memory of a single computer.
- ... the dimensions are too high, that we cannot build a model.
- ... the calculations needs more than a week to be processed.

To recap. Big Data is...

When...

- ... data that will not fit in main memory,
- ... data cannot be handled by traditional data processing techniques
- ... a calculation needs more than a single computer to process.
- ... a dataset is too large to fit into the memory of a single computer.
- ... the dimensions are too high, that we cannot build a model.
- ... the calculations needs more than a week to be processed.

To recap. Big Data is...

When...

- ... data that will not fit in main memory,
- ... data cannot be handled by traditional data processing techniques
- ... a calculation needs more than a single computer to process.
- ... a dataset is too large to fit into the memory of a single computer.
- ... the dimensions are too high, that we cannot build a model.
- ... the calculations needs more than a week to be processed.

To recap. Big Data is...

When...

- ... data that will not fit in main memory,
- ... data cannot be handled by traditional data processing techniques
- ... a calculation needs more than a single computer to process.
- ... a dataset is too large to fit into the memory of a single computer.
- ... the dimensions are too high, that we cannot build a model.
- ... the calculations needs more than a week to be processed.

To recap. Big Data is...

When...

- ... data that will not fit in main memory,
- ... data cannot be handled by traditional data processing techniques
- ... a calculation needs more than a single computer to process.
- ... a dataset is too large to fit into the memory of a single computer.
- ... the dimensions are too high, that we cannot build a model.
- ... the calculations needs more than a week to be processed.

With the goal of...

Technologies for Big Data

Frameworks

- **Distributed Computing**: Hadoop, Spark, Flink, ...
- **Cloud Computing**: AWS, Azure, Google Cloud, ...
- **Parallel Computing**: MPI, OpenMP, ...
- **Data Storage**: HDFS, S3, ...
- **Data Visualization**: Tableau, PowerBI, ...
- **Machine Learning**: TensorFlow, PyTorch, ...
- **Streaming**: Kafka, Kinesis, ...

Analytics and Algorithms

- **Data Mining**: Clustering, Association, ...
- **Similarity Search**: LSH, ...
- **Hypothesis Testing**: T-Test, ANOVA, ...
- **Transformers**: PCA, LDA, ...
- **Recommender Systems**: Collaborative Filtering, Multi-Armed Bandit, ...
- **Link Analysis**: PageRank, HITS, ...

Hadoop

What is Hadoop?

- Open-source software framework for distributed storage and processing of large datasets across clusters of computers.
- Designed to scale up from a single computer to thousands of clustered computers, each offering local computation and storage.

What is Hadoop?

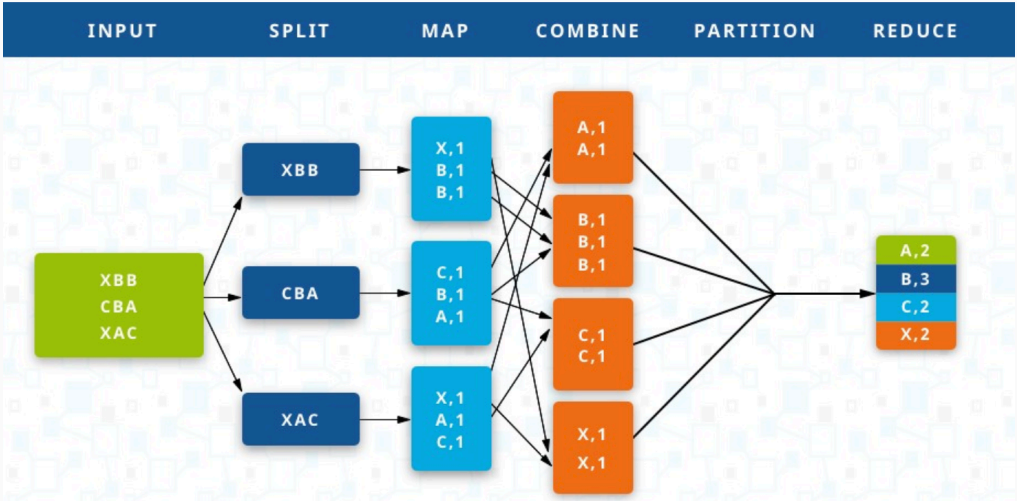
- Open-source software framework for distributed storage and processing of large datasets across clusters of computers.
- Designed to scale up from a single computer to thousands of clustered computers, each offering local computation and storage.

Main Components

- **Hadoop Distributed File System (HDFS)**: A distributed file system that provides high-throughput access to application data.
- **Yet Another Resource Negotiator (YARN)**: A resource management platform responsible for managing resources in a cluster and scheduling user applications.
- **MapReduce**: A programming model for processing and generating large datasets that is parallelizable across a distributed cluster.

What is MapReduce?

MapReduce is a *data processing* job that divides the input data into chunks, which are processed by the map function and then reduced by grouping similar sets of data



Examples of MapReduce

Could you think of other examples where MapReduce could be used?

Examples of MapReduce

Could you think of other examples where MapReduce could be used?

- **Genomic Data Processing**: MapReduce can be used to process and analyze large genomic datasets, such as those generated by genome sequencing. This can help identify genetic variations and their associations with different diseases.

Examples of MapReduce

Could you think of other examples where MapReduce could be used?

- **Genomic Data Processing**: MapReduce can be used to process and analyze large genomic datasets, such as those generated by genome sequencing. This can help identify genetic variations and their associations with different diseases.
- **Electronic Health Records (EHR) Analysis**: EHRs contain a wealth of information about patients' health histories. MapReduce can be used to analyze EHRs on a large scale to identify patterns and trends, which can inform healthcare policies and practices.

Examples of MapReduce

Could you think of other examples where MapReduce could be used?

- **Genomic Data Processing**: MapReduce can be used to process and analyze large genomic datasets, such as those generated by genome sequencing. This can help identify genetic variations and their associations with different diseases.
- **Electronic Health Records (EHR) Analysis**: EHRs contain a wealth of information about patients' health histories. MapReduce can be used to analyze EHRs on a large scale to identify patterns and trends, which can inform healthcare policies and practices.
- **Medical Imaging Analysis**: MapReduce can be used to process and analyze large amounts of medical imaging data to assist in diagnosis and treatment planning.

Examples of MapReduce

Could you think of other examples where MapReduce could be used?

- **Genomic Data Processing**: MapReduce can be used to process and analyze large genomic datasets, such as those generated by genome sequencing. This can help identify genetic variations and their associations with different diseases.
- **Electronic Health Records (EHR) Analysis**: EHRs contain a wealth of information about patients' health histories. MapReduce can be used to analyze EHRs on a large scale to identify patterns and trends, which can inform healthcare policies and practices.
- **Medical Imaging Analysis**: MapReduce can be used to process and analyze large amounts of medical imaging data to assist in diagnosis and treatment planning.
- **Public Health Surveillance**: MapReduce can be used to analyze large datasets from various sources (like social media, hospital records, etc.) for public health surveillance. This can help identify and track the spread of diseases and other health-related trends.

... and many more!

Map Reduce applied to Drugs Interaction

Is there a statistically significant interaction between two drugs?

Map Reduce applied to Drugs Interaction

Is there a statistically significant interaction between two drugs?

- **Mapper**: The input is a set of drug records. Each record could contain information about a patient, the drugs they are taking, and any observed reactions. $\Rightarrow$ The Map function processes each record independently and emits pairs of drugs that were taken together by a patient. For example, **if a patient took Drug A and Drug B, the Map function would emit the pair (Drug A, Drug B)**.

Map Reduce applied to Drugs Interaction

Is there a statistically significant interaction between two drugs?

- **Mapper**: The input is a set of drug records. Each record could contain information about a patient, the drugs they are taking, and any observed reactions. $\Rightarrow$ The Map function processes each record independently and emits pairs of drugs that were taken together by a patient. For example, **if a patient took Drug A and Drug B, the Map function would emit the pair (Drug A, Drug B)**.
- **Shuffle and Sort Phase**: This is an intermediate phase managed by the MapReduce framework. All the emitted pairs from the Map phase are collected, sorted, and grouped. So, all occurrences of a particular pair of drugs are brought together.

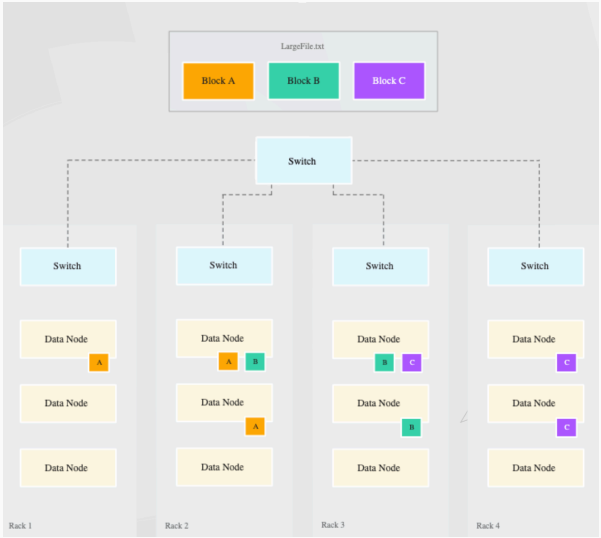
Map Reduce applied to Drugs Interaction

Is there a statistically significant interaction between two drugs?

- **Mapper**: The input is a set of drug records. Each record could contain information about a patient, the drugs they are taking, and any observed reactions. ⇒ The Map function processes each record independently and emits pairs of drugs that were taken together by a patient. For example, **if a patient took Drug A and Drug B, the Map function would emit the pair (Drug A, Drug B)**.
- **Shuffle and Sort Phase**: This is an intermediate phase managed by the MapReduce framework. All the emitted pairs from the Map phase are collected, sorted, and grouped. So, all occurrences of a particular pair of drugs are brought together.
- **Reducer**: The Reduce function takes in a pair of drugs and the list of all occurrences of that pair. It then analyzes these occurrences to determine if there's a statistically significant interaction between the two drugs.

What is HFDS?

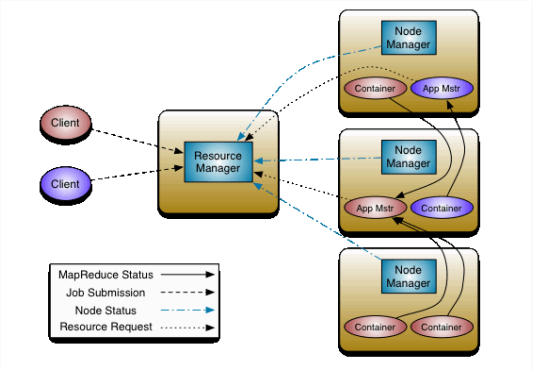
High-throughput and Fault-tolerant Distributed File System



YARN (Yet Another Resource Negotiator)

YaRN is a resource management layer in the Apache Hadoop ecosystem that provides a central platform for managing computing resources and scheduling tasks across a distributed cluster of machines.

- **ResourceManager**: Manages the computing resources across the cluster, including memory, CPU, and disk space.
- **ApplicationMaster**: Manages the execution of a specific application or job.



Related tools and frameworks

- **Apache Hive**: A data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis.

Related tools and frameworks

- **Apache Hive**: A data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis.
- **Apache Sqoop**: A tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.

Related tools and frameworks

- **Apache Hive**: A data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis.
- **Apache Sqoop**: A tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.
- **Apache Impala**: An open-source massively parallel processing SQL query engine for data stored in Hadoop.

Related tools and frameworks

- **Apache Hive**: A data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis.
- **Apache Sqoop**: A tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.
- **Apache Impala**: An open-source massively parallel processing SQL query engine for data stored in Hadoop.
- **Apache HBase**: A distributed, scalable, non-relational database.

Related tools and frameworks

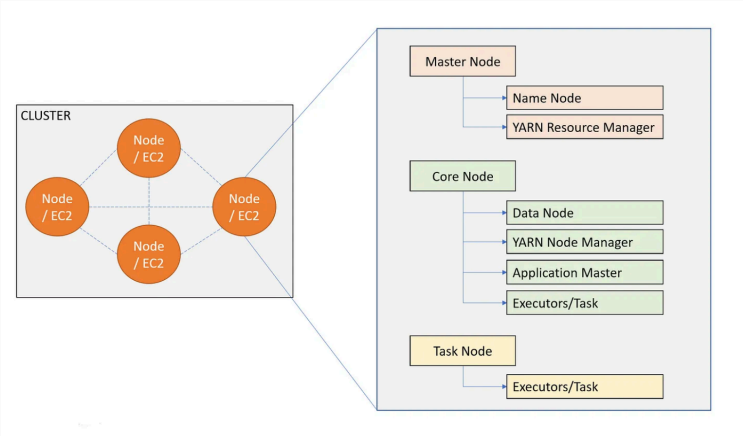
- **Apache Hive**: A data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis.
- **Apache Sqoop**: A tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.
- **Apache Impala**: An open-source massively parallel processing SQL query engine for data stored in Hadoop.
- **Apache HBase**: A distributed, scalable, non-relational database.
- **Apache Spark**: A fast and general-purpose cluster computing system that provides APIs in Java, Scala, Python, and R.

Hadoop in AWS

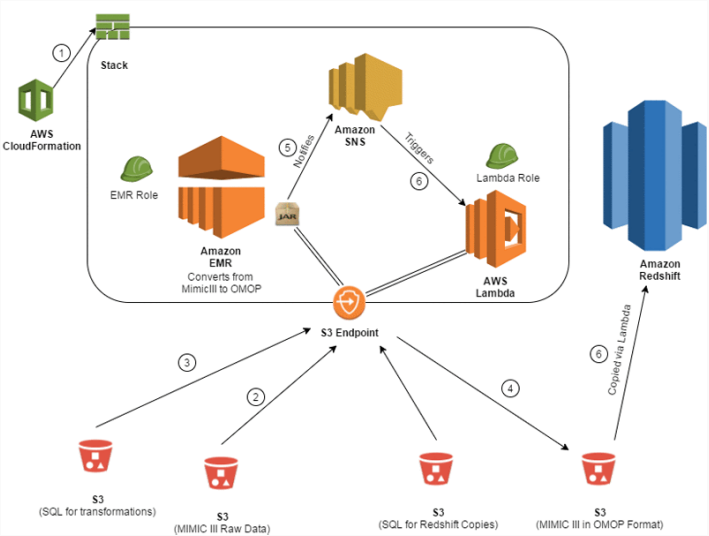
In AWS, there is a service called **Amazon EMR** that provides a managed Hadoop framework that makes it easy, fast, and cost-effective to process vast amounts of data across dynamically scalable Amazon EC2 instances.

Terms

- **Cluster**: A group of EC2 instances that work together to process and analyze data.
- **Master Node**: Manages the cluster and coordinates the distribution of tasks.
- **Core Nodes**: Store data and run tasks in parallel.
- **Task Nodes**: Run tasks in parallel.



Example 1: Data Warehousing in AWS using EMR



HandsOn: 1000 Genomes - Population Distribution

1000 Genomes Project

- The 1000 Genomes Project is an international research effort to establish a detailed catalog of human genetic variation.

1000 Genomes Project

- The 1000 Genomes Project is an international research effort to establish a detailed catalog of human genetic variation.
- The project's data is freely available to both the scientific community and the general public on **1000 Genomes website**.

1000 Genomes Project

- The 1000 Genomes Project is an international research effort to establish a detailed catalog of human genetic variation.
- The project's data is freely available to both the scientific community and the general public on **1000 Genomes website**.
- This data includes genomic sequences, genetic variants, and functional annotations for a large number of individuals from diverse populations around the world.

1000 Genomes Project

- The 1000 Genomes Project is an international research effort to establish a detailed catalog of human genetic variation.
- The project's data is freely available to both the scientific community and the general public on **1000 Genomes website**.
- This data includes genomic sequences, genetic variants, and functional annotations for a large number of individuals from diverse populations around the world.
- For this experiment, we will use a small sample file called **integrated_call_male_samples_v3.20130502.ALL.panel**. This file contains information about the *samples in the dataset*, which are grouped into populations and super populations based on their geographic origin.

Case Study: Population Distribution

Objective

The study aims to determine the distribution of samples across different populations in the 1000 Genomes dataset.

Case Study: Population Distribution

Objective

The study aims to determine the distribution of samples across different populations in the 1000 Genomes dataset.

Sample Information

The sample contains information about the samples in the dataset. The fields in the file are as follows:

- **Sample**: The name of the sample.
- **Population**: The population to which the sample belongs.
- **Super Population**: The super population to which the sample belongs.

Buckets for Input and Output

1. Go to the Amazon S3 console at S3 AWS.
2. Choose **Create bucket**.
3. Enter a name for your bucket and choose the region where you want to create the bucket. *This should be the same region where your EMR cluster is located*.
4. Choose **Create**.
5. Repeat the process to create a second bucket for the output of your MapReduce job.

For example, you could name your buckets *hpdc–1000genomes-input* and *hpdc–1000genomes-output*.

MapReduce: Code structure in Python

```
import json
import sys

def map_function(record):
    # TODO: Implement the map function

def reduce_function(key, values):
    # TODO: Implement the reduce function

# Read the input from standard input
for line in sys.stdin:
 # Skip the header line
    if not line.startswith('#'):
        # Call the map function
        map_function(line.strip())
```

MapReduce: Mapper

```python
#!/usr/bin/env python3
import sys

def map_function(record):
    fields = record.split('\t')
    population = fields[1]
    super_population = fields[2]
    print(f'{population}\t{super_population}\t1')

for line in sys.stdin:
    if not line.startswith('#'):
        map_function(line.strip())
```

MapReduce: Reducer

```python
import sys
from itertools import groupby
from operator import itemgetter

def read_mapper_output(file, separator='\t'):
    for line in file:
        yield line.rstrip().split(separator, 2)

def main(separator='\t'):
    data = read_mapper_output(sys.stdin, separator=separator)
    for current_key, group in groupby(data, itemgetter(0)):
        total_count = sum(int(count) for _, _, count in group)
        print(f"{current_key}{separator}{total_count}")

if __name__ == "__main__":
    main()
```

Upload the MapReduce script to S3

1. Go to the Amazon S3 console at S3 AWS.
2. Choose the bucket you created for the input data.
3. Choose **Upload**.
4. Choose **Add files** and select the MapReduce script (**map.py** and **reduce.py**).
5. Choose **Upload**.
6. Create a folder called **experiment1** in the bucket.
7. Choose **Add files** in the **experiment1** folder and select the data (**integrated_call.ALL.panel**).
8. Choose **Upload**.

Building an EMR Cluster

Go to the Amazon EMR console at EMR AWs. Select **Create cluster**.

Name and Applications

- **Name**: hpdc–EMR
- **EMR Release**: emr-7.0.0
- **Applications**: Custom - **Select only Hadoop**

Scaling and Provisioning

- Select **Set cluster size manually**.
- **Provisioning**: 1 core and 1 task

Network

- Select the subnet in region: *us-east-1a*

Cluster Configuration

- Select **Uniform instance groups**.
- Number of instances: 3 (primary, core, and task)
- Instance type: m5.xlarge

Identity and Access Management

- Choose **EMR_DefaultRole** as service role.
- Choose **EMR_EC2_DefaultRole** as EC2 instance profile.

Security

- **EC2 key pair**: key pair you have created.

Running the MapReduce job

1. Go to the Amazon EMR console at EMR AWS.
2. Choose the cluster you created.
3. Choose **Add step**.
4. Choose **Streaming program**.
5. Enter a name: **hpdc–experiment1**.
6. Enter the following information:
    - **Input S3 location**: s3:hpdc–1000genomes-input/experiment1/
    - **Output S3 location**: s3://hpdc–1000genomes-output/experiment1
    - **Mapper**: s3://hpdc–1000genomes-input/map.py
    - **Reducer**: s3://hpdc–1000genomes-input/reduce.py
7. Choose **Add**.

Visualizing the results

1. Go to the Amazon QuickSight console at QuickSight AWS.
2. Choose **Create new dataset**.
3. Choose **New dataset**.
4. Choose **AWS data source**.
5. Choose **S3**.
6. Enter the path to the output of your MapReduce job in the S3 bucket field.
7. Choose **Create data source**.
8. Choose **Edit/Preview data**.
9. Choose **Visualize**.
10. Choose the **Bar chart** icon.
11. Drag the population field to the X-axis field well.
12. Drag the total field to the Value field well.
13. Choose **Save & visualize**.

Visualizing the results in Jupyter (I)

```
# Install boto3, matplotlib and pandas
# aws configure

import boto3
import matplotlib.pyplot as plt
import pandas as pd

s3 = boto3.client('s3',region_name='us-east-1')

# Define the S3 bucket name and prefix
bucket_name = 'hpdc-<yourname>-1000genomes-output'
prefix = 'experiment1'

# Get a list of all object keys in the bucket with the specified prefix
response = s3.list_objects_v2(Bucket=bucket_name, Prefix=prefix)
object_keys = [obj['Key'] for obj in response['Contents']]
```

Visualizing the results in Jupyter (II)

```
# Initialize an empty list to store the data
data = []
# Loop through the object keys and read the data into a list
for obj_key in object_keys:
    obj = s3.get_object(Bucket=bucket_name, Key=obj_key)
    lines = obj['Body'].read().decode().split('\n')
    for line in lines:
        if line:
            # Parse the line into population and count columns
            cols = line.split()
            population = cols[0]
            count = int(cols[1])
            data.append([population, count])
# Create a DataFrame from the data
df = pd.DataFrame(data, columns=['population', 'count'])
```

Visualizing the results in Jupyter (III)

```
# Group the data by the population column and sum the count column
grouped_df = df.groupby('population')['count'].sum().reset_index()

# Create a bar plot
plt.figure(figsize=(10,6))
plt.bar(grouped_df['population'], grouped_df['count'])
plt.xlabel('Population')
plt.ylabel('Count')
plt.title('MapReduce Results')
plt.xticks(rotation=45)

# Show the plot
plt.show()
```

Conclusions

Conclusions

- Big Data is a term used to describe large and complex datasets that are difficult to process using traditional data processing techniques.

Conclusions

- Big Data is a term used to describe large and complex datasets that are difficult to process using traditional data processing techniques.
- Big Data presents several challenges, but also opportunities for extracting valuable insights and knowledge from large volumes of data.

Conclusions

- Big Data is a term used to describe large and complex datasets that are difficult to process using traditional data processing techniques.
- Big Data presents several challenges, but also opportunities for extracting valuable insights and knowledge from large volumes of data.
- Hadoop and MapReduce are two key technologies for processing and analyzing big data using parallel and distributed computing.

Conclusions

- Big Data is a term used to describe large and complex datasets that are difficult to process using traditional data processing techniques.
- Big Data presents several challenges, but also opportunities for extracting valuable insights and knowledge from large volumes of data.
- Hadoop and MapReduce are two key technologies for processing and analyzing big data using parallel and distributed computing.
- Amazon EMR is a managed framework that makes it easy, fast to deploy a Big Data infrastructure in AWS.

That’s all

Thanks for your attention!

Questions?