# COMP 1702
# Big Data: Coursework 2

Khassan Nazhem
001151725
Dr. Hai Huang
Dr. Mariusz Pelc
March 29, 2021

# Table of Contents



# List of Figures

# Part A:

The algorithm adopted for such a problem is similar to the word count algorithm, however in this case we need to count a specific field which is the names of the conferences and check how many times each name is repeated. Since we are assuming there are no duplicates, then each conference record count refers to a paper. Thus, the total number of repetitions (count) of a conference is the number of papers. To output the number of papers for each conference, we should follow the Map-Reduce process. A map-reduced algorithm is applied to divide and conquer, where data is broken over computing and parallel processing. Hadoop distributed file system (HDFS) divides the files into blocks with a size of 128MB; this allows higher efficiency for the processing of large files and receives the whole output instantly. Through looping over every line, the conference names fields are retrieved based on the bar "|" delimiter using line splitter, moving on to saving element number 2 from the split list conferences [2]. The process initializes the data as a set of <key, value> pairs for processing. In our case, the function breaks the input into a mapped <conference, count (1)> pairs. While the mapper task is running, the data is relocated to the next node to receive optimum performance. At shuffling level, the mapped <conference, count> pair values are gathered with the same key (conference name) and then put into the same reducer machine. The reducing process starts adding the counts of all the collected <conference, count>. The program finally returns the result including the conference name and its summed count. Map Reduced functions rely on replicas that are created and stored in local nodes. In addition, other replicas are stored in the rack so that Hadoop can avoid losing any failure points or losing data.

## Map-Reduce Algorithm:

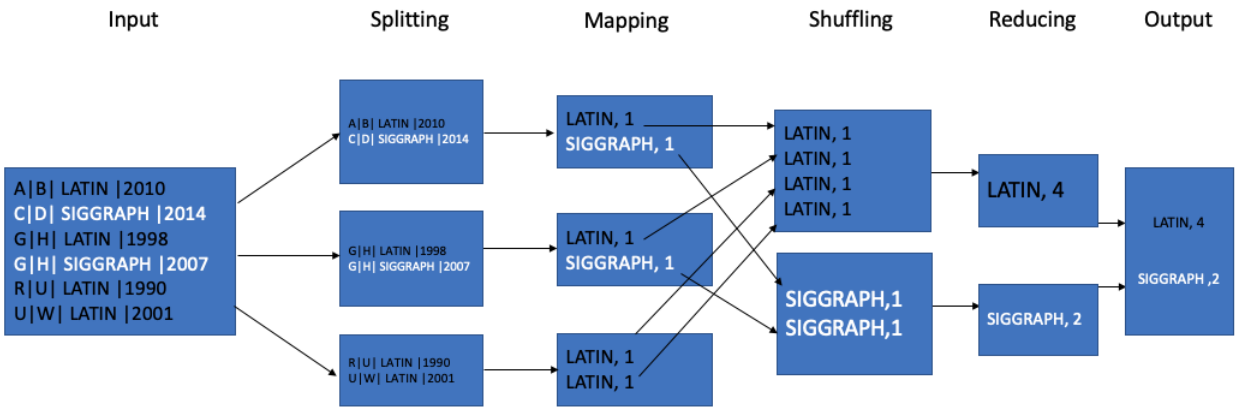


*Figure 1: MapReduce Algorithm*

## Code

```
public static class Map extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
  private final static IntWritable one = new IntWritable(1);
  private Text word = new Text();

  public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
    String line = value.toString();
    BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

    while((line = reader.readLine()) != null){
        String[] conferences = line.split("|");
        String conference = conferences[2];
        word.set(conference);
        output.collect(word, one);
    }

  }
}

public static class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {
  public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
    int sum_conference = 0;
    while (values.hasNext()){
      sum_conference += values.next().get();
    }
    output.collect(key, new IntWritable(sum_conference));
  }
}
```

*Figure 2: MapReduce Code*

## Output Screenshot: 1st 10 lines (Conference name – Paper count)

International Conference on Computational Science (2) 23
LATIN 5
SIGGRAPH 32
Applied Informatics 14
ICIP 52
ICPP Workshops 20
VLSI Design 34
CCGRID 16
ECAL 10
PRIMA 4
IC-AI 20

*Figure 3: MapReduce Output*

# Part B:

## Task 1:

### Data Sources:

ArgBIG's big data architecture requires to be highly available, scalable, and accessible from everywhere due to its huge load of data. Normally, the big data architecture has to handle and process data from different sources and formats which are extracted in both batches and in real-time. Afterward, the data is stored and then queried for various analytical studies. Our resources in this case include structured, semi-structured, and unstructured data coming from soil sensors, agricultural product prices, historical weather data, online feed, satellite images, and drone videos.

The soil sensors' job is to constantly monitor and collect data by providing real-time information on soil moisture, soil temperature, water intake, and other metrics to track trends and predict irrigation needs. Moreover, historical weather data is a set of data that includes past weather conditions in a particular region. These records can supply multiple weather metrics such as temperature, wind direction, rainfall, speed, and humidity. Historical weather data can be as recent as a week ago. However, the longer the duration, the more detailed the record is, making it more valuable. This data can be integrated with the satellite images and videos since they confirm the atmospheric behavior. Additionally, the company can extract a lot of information from social media like Twitter, Instagram, and Facebook. Social media data creates trends and highlights that help in carrying out business and public relations research.

### Data Extraction:

Ideally, the company relies on getting results in real-time and combine them with other raw data from other data sources. Hence, it is recommended to use a system that goes through two paths. The first one is the cold path, which stores all historical weather, satellite images, and drone videos in data lakes. The second path, which is the hot path, is responsible to take data from the stream ingestion with the most recent incremental data. Furthermore, it analyzes information in real-time and then either stores them in the data lakes. So, when these two paths are interlayered together, the data going through them is then stored in the data lake to follow up with the analytics application. With that being said, if the user requires to display data on the spot – which can be less accurate - the data will be claimed from the streaming process (hot path). Else, the user can use the data stored (cold path) in the data lake for more accurate data.

### Data Cleaning:

The goal of data cleaning is to make sure that all the data sets are standardized and uniform. This allows data analytics tools to access and manipulate the data for each query. Hence, data cleaning consists of identifying and removing, or modifying inaccurate records from a dataset and then remodeling the data. For example, the weather data should follow a constant formatting that includes issue dates, issued data, and measures. Additionally, soil sensors datasheet should include the range of data issued, the units used, the source of the data, and the issuing date. A great tool that offers such functionalities is AWS Glue Data Brew.

### Data Storage:

Due to the fact that the data sources of this project have a wide variety of types, the adopted storage should be able to store large amounts of data that can be of any format in a scalable and fault-tolerant routine. Initially, the storage should operate over the raw loaded data like weather information and corresponding media would effectively be stored in a graph database like Neo4J. Also, the column family databases should stay up to date with all of the new data (soil moisture, soil temperature, water intake) and versions that were created by the real-time streaming. Raw and original data is captured and comes in handy in exploration and early reporting. This is the reason as to why data catalog is used to provide a query-able interface of all data stored and their relevance to the company's application.

### Batch Processing:

Through batch processing, all the data coming from different sources like historical weather data, sensors, satellite images, and drone videos are stored in data lakes which then performs batch jobs filter. The jobs that are involved in batch processing focus on reading the data sources, processing them, and then creating new files to output the retrieved data for advanced analytics. Batch processing fits best with MapReduce jobs since in situations the main goal is to indulge large datasets rather than trying to process the data in a fast pace.

### Real Time Message Ingestion:

Data coming from soil sensors and streaming agricultural product prices go through sourcing, manipulating and loading as soon as it's created. However, it is more expensive since it requires constant monitoring, but it is appropriate for analytics that require continually refreshed data. In this case, tools like Apache Kafka or Amazon's Kinesis would perform greatly since we care about log monitoring for the existing sensors and monitors to perform analysis for the plants behavior when responding to weather or agricultural stimulus, or customer behavior and logs through social media interactions.

### Analytical Data Store:

After cleaning the data, it is either stored in relational data warehouses to serve analytical queries or in low-latency NoSQL databases like Hive to ensure faster queries and more efficient usage. To support data querying through analytical tools, the data should be structured and processed through the analytical data store component. These tools provide insight into the data and support interactive exploration for data scientists and analysts. The output will be a great asset to maintain and monitor the quality of yield through understanding the implications of the weather, soil products and pests on plant farming. Furthermore, the performance of buy and sell demand for agricultural supplies is easily detected.

## Task 2:

Since the company needs to store a collection of data for plants, crops, pests, and their relationships, then a graph database like Neo4j is the most suitable database for such requirements. A graph database mainly focuses on relationships between data as much as it focuses on storing the data itself. This data value is stored as a label having properties it connects with other related entities according to the relationship, thus it doesn't require any pre-defined models or schemas. Through these relationships, the data is stored, processed, and queried efficiently. Hence, graph databases permit time-efficient operations whilst traversing millions of connections. Through specifying a set of starting points and a traversing pattern, graph databases explore all the data neighbors around those starting points to collect and aggregate information.

Neo4J uses Cypher; Cypher is like SQL but used for graphs. It comprehends many expressions and queries that are familiar like CREATE, WHERE ORDER BY, SKIP and so on. The SELECT is annotated using a special clause MATCH: it is used in to match the given nodes and patterns in the database. Node entities are presented with parentheses as such: (p: Plant), whilst relationships are presented using arrows as such ->. The relationship-types are added in square brackets - [: Causes]->. Therefore, a full Cypher query that retrieves all the diseases that affect corn and created by zin deficiency would look like the following:

```
MATCH p = (s: Symptom)- [: Causes]-> (d: Disease) - [: Infects]-> (p: Plant)
WHERE s.name = "Zinc Deficiency" AND p.name = "Corn"
RETURN p
```

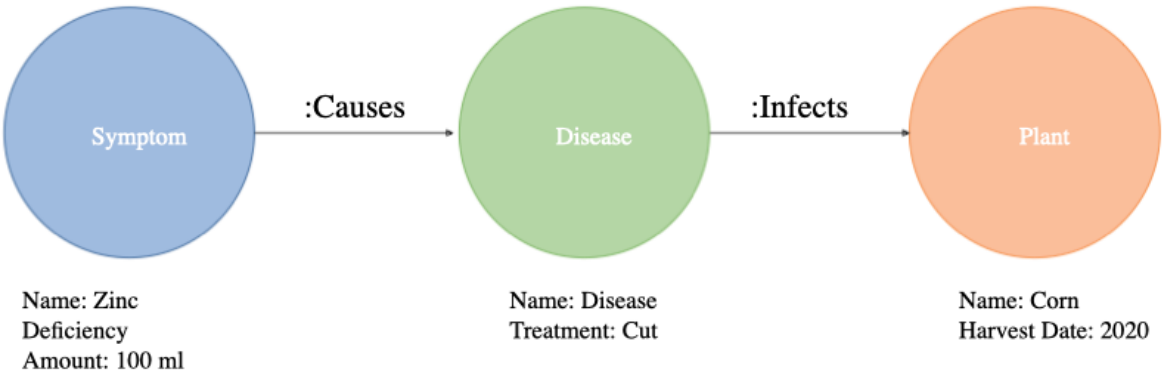


*Figure 4: Neo4J Map*

## Task 3:

AgrBIG company is looking for a solution that can replace Map-reduce but also be a SQL-like language to perform big data analysis tasks. Apache Hive fits perfectly for this scenario. Hive is a data warehouse infrastructure tool that is used within Hadoop to process structured data. It was created to help non-programmers who are more familiar with SQL to apply processing on collections of data by using a SQL-like interface called HiveQL. So, instead of using MapReduce, programmers can use HiveQL queries; Hive transforms these queries into MapReduce that runs on Hadoop's YARN. The main advantage of Apache Hive is the ability to query, summarize, and analyze data. It provides easy to use functions so that the process for developers when working with various data formats and complex analytical processing can be easier. In addition, it provides high and flexible performance when adding more data or increasing the number of nodes in the cluster. Hence, compared to the other query languages, Hive has a much faster response time when functioning on huge data sets.

## Task 4:

The company's goal is to receive real-time data processing that can allow the users like farmers, research laboratories, and administration to take a crucial business decision at the moment. Even though Hadoop's MapReduce is known as a software that can process vast amounts of data in-parallel on large clusters, it is better to find an alternative. Apache Spark is able to perform services like soil moisture prediction, weather predictions or agricultural products price predictions. This is especially important for AgrBIG since it integrated a lot of IoT devices, machine-to-machine communications, and pre-existing data. Thus, knowing the nature of tasks to be achieved puts MapReduce in the second place as is not up to the challenge with the Big Data demands when compared to Apache Spark. That is because MapReduce does not support real-time processing which is the main demand for the company. At the same time, the company is looking for adapting scalable components for its architecture, however building a MapReduce program can be very time-consuming and exhausting since it is considered a low-level language.

Correspondingly, Spark is fully compatible with Hadoop Distributed File System and it can be a suitable replacement for MapReduce. Spark is able to function through diverse data sources and types. Regardless of MapReduce being a well-organized data processing component of Hadoop, its full performance comes in handy for batch processing. Therefore, it is MapReduce's job to linearly process enormous datasets while Spark focuses on delivering high performance through its in-memory data processing. In addition, it supports real-time analytics, the ability to process graphs, and machine learning. Machine learning algorithms are a crucial component to process weather forecasts and agricultural components data. Spark has a built-in library MLlib for machine learning, whereas Hadoop requires a third-party to provide such processing. These algorithms have a role in defining real-time company needs and decisions through giving immediate insights and in-memory processing. Also, a chunk of the dataset focuses on the relationship between crops, pests, diseases, and their symptoms which makes Spark very compatible for graph processing through its nonlinear model.

## Task 5:

Through cloud computing, the company will receive enhanced computing services, storage, analytics, and intelligence without worrying about scalability, performance, and high availability globally. Hybrid clouds are ideal for large-sized businesses like AgrBIG, providing a more tailored IT solution that meets all the requirements and is linked to the private Neo4j database. Through Amazon AWS Hybrid Clouds, AgrBIG will be fully equipped with services and infrastructure that meets their demands. Precision agriculture (PA) requires high technology sensors and analysis tools; the usage of cloud computers can provide faster performance since it doesn't put too much load on the memory. Besides, cloud computing offers virtually limitless storage knowing that AgrBIG's data volume is 300 petabytes. Finally, the software will be available globally since the cloud has universal document access that can instantly retrieve the latest version.

The data initiated can be stored in a distributed file system like Amazon's S3 because it can hold high volumes in various formats assisted with batch processing through Hadoop MapReduce since it provides parallel processing of large data. Furthermore, to apply a continuous and real-time data Ingestion, Amazon Kinesis can be used since it is highly scalable, durable, and reliable. It can be scaled rapidly and easily without experiencing any interruption. Moreover, Kinesis is a fully managed service that supports stream processing for real-time data analysis. Alongside is Apache Spark that delivers in-memory data processing to support real-time analytics, graph processing and machine learning. Using some existing software or services provided by Amazon like AWS Glue Data Brew would enable the user to remodel that data without writing any code. There exist more than 250 ready transformations to automate data preparation tasks. These tasks are defined by highlighting irregularities, standardizing the data formats, and visualizing clean raw data. The analytical data store is hosted by Apache Hive, offering a variety of functions that can ease complex analytical processing. Integrated with Quick Sight, users in the company will be able to receive scalable machine learning-powered business intelligence (BI), in addition to creating and publishing interactive data analytical dashboards.

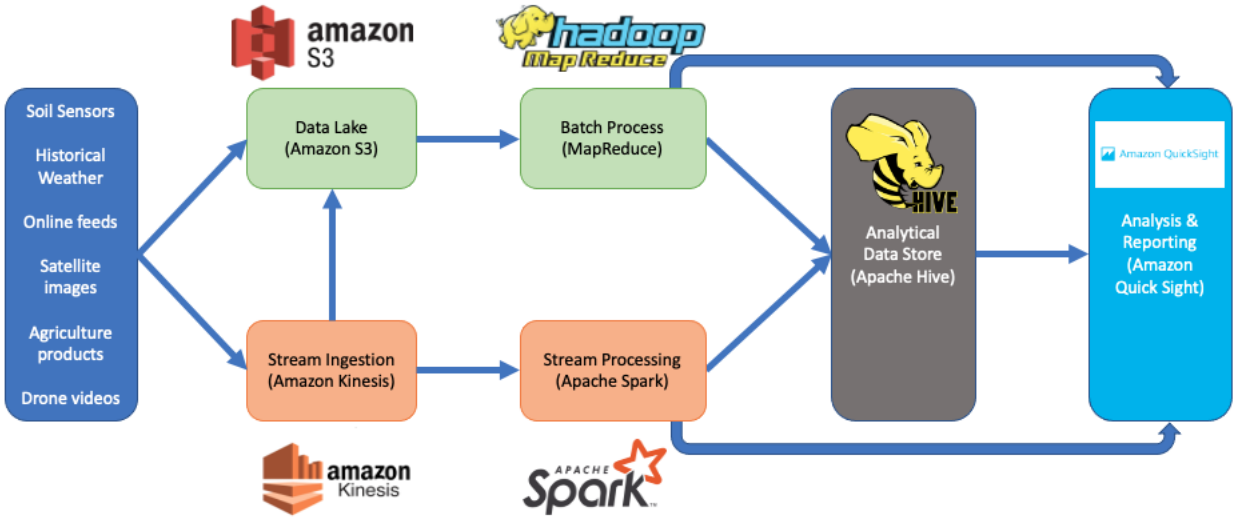


*Figure 5: Cloud solutions*

# Module Evaluation:
(required by department)

Through this module, I was able to gain knowledge on the foundation of Big Data and the limitations opposed by the tools used to manipulate Big Data. In addition, the first coursework presented efficient tasks to explain Hadoop's MapReduce, NoSQL databases and RDBMS. Furthermore, the second coursework successfully presented a challenging atmosphere to use the learnt skills in order to develop a solution and a Big Data architecture that requires manipulating Big Data. Such skills include analyzing large data, integrating Big Data technologies and choosing the suitable deployment system. These were reflected through using Hadoop to store, process, and analyze large collections of data through using its components like HDFS, MapReduce, Hive and Spark. Additionally, the Big Data architecture implemented included full understanding of Amazon Web Services, Microsoft Azure Services and the following components: data sources, data extraction and cleaning, data storage, batch processing, real time message ingestion, and analytical data store.