COMP 1702

BIG DATA

STUDENT NAME: SAI SANDEEP CHITTIBOINA

STUDENT ID: 001203994

COURSE LEADER: DR.HAI HUANG

TASK A.1

The three main characteristics of Big Data are:

- The Volume of the data

Nowadays zettabytes (ZB) of data was being produced each year, it's no longer able to store that much amount of data by individual enterprises. To hold that big data we can use the right technology to analyze and get a better understanding of the insights of companies & your customers & market place.
This large amount of data helps the companies to generate more accurate insights with better accuracy and help to make more beneficial decisions and to avoid risks

- The variety of the data

With the increase in the technology like sensors and smart devices, the data that we are receiving is getting more complex that including raw, semi-structured, unstructured and structured data these are some of the sources from where we are getting the data i.e. web pages, search indexes, social media, e-mails, documents, articles etc

The variety in data types requires frequent distinct processing techniques and special algorithms to acquire the required format of data.

- The velocity of the data

For easy understanding, the velocity of data means how quick the data is arriving and getting stored and distributed in the database the rate of the velocity of the data will be based on the number of sensors used in an IoT device or the number of posts in Facebook posted in a day, or the number of Twitter tweets in a day this all are the factors to increase the velocity of data.

TASK A.2

Comparison between Hadoop and Relational database system

The difference between Hadoop and relational databases is given below in the table form:

Relational database	Hadoop
Structured data is processed mostly here.	Both unstructured and structured data are processed here.
There is no delay in response.	There is some delay in response.
It has a static type of data schema	It has a dynamic type of data schema
This is generally a traditional database	It can store any kind of data structured and unstructured.
Data is more accurate when compared to Hadoop	Data is less accurate when compared to relational database
This is a paid database	This is free of cost is an open-source software
The normalization of the data required	The normalization of data is not required
Read the data fast	Reads and writes the data fast
It has vertically scalability.	It has horizontally scalability.
It's used in high-end servers.	It's used in commodity hardware

Hadoop uses two main components HDFS and MapReduce. Hadoop can store a large amount of data in layers. The map-reduce is a base model which can covert a large amount of data into small portions into different data blocks.

Anyhow the relational database is a structured database which stores the data in rows & columns, it uses SQL which helps to update and access the loaded data in various tables but this cannot be used for storing large amounts of the data.

One of the important points in comparison between these is RDBMS retrieves the information very quickly when compared with Hadoop but it has higher outputs when compared to RDBMS in this was Hadoop take a significant advantage over the relational database, Hadoop is completely free and open source when it comes to costs, and RDBMS has licensed software you have to pay for.

TASK B.1

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class Average_grades {

    public static class Text_mapper
        extends Mapper<Object, Text, IntWritable>
{       private Text_module = new Text();
         private Float_writable grade = new Float_writable();
public void map(Object key, Text value, Context context)
throws IOException, InterruptedException
        {
          String[ ] lines = value.split("\\r?\\n");
for (int i = 1; i < lines.length; i++) {
row = lines[i];
            String[ ] fields = row.split(",");
module.set(fields[4]);              grade.set(fields[5]);
            context.write(module, grade);
          }
      }
    }
}
public static class Grade_reducer
    extends Reducer<Text, IntWritable, Text, IntWritable>
{     private IntWritable result = new IntWritable();
    public void reduce(Text key, Iterable<IntWritable> values, Context context)
throws IOException, InterruptedException
    {
      int count = values.length;
float sum = 0;
```

```
      for (IntWritable val : values) {
sum += val.get();
      }
      result.set(sum/count);
      context.write(key, result);
    }
  }
  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
Job job = Job.getInstance(conf, "word count");
job.setJarByClass(Average_grades.class);
job.setMapperClass(Text_mapper.class);
job.setCombinerClass(Grade_reducer.class);
job.setReducerClass(Grade_reducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

TASK B.2

My algorithm code consists of 3 classes namely Mapper, Combiner and Reducer.
Below are the functions performed by all three classes.

Average_grades.class; Grade_reducer.class; Text_mapper.class

• Mapper: First the algorithm splits the entire data into smaller parts and passes it on to the mapper function in form of <key, value> pairs. This function then takes these pairs to process intermediate <key, value> pairs. In our situation, each mapper takes pair of <Module, Grade> and then returns the partial count of the numbers in each following grade with each module in each subset.

• Combiner: The combiner then takes these pairs <key, value> and aggregates the result for each mapper. For our solution, the combiner will take the partial counts and the partial grade sums and returns the pairs of grade sum and counts for each 'Module'.

• Reducer: The job of the reducer is to shuffle and sort the intermediate results and then combine the set of results into a final result. In our solution, the reducer takes the partial grade sum and counts for each module and finally returns the average grade of each module.

The efficiency of the algorithm:
my algorithm code implements combiners to make it efficient as they store the intermediate results and pass it on to the reducers. Without combiners, the count for in-between <key, value> pairs would be the same as the number of input <key, value> pairs and the slider will still give the right result, which means it does not affect the accuracy of the algorithm but increases the workload and thus efficiency.. As our algorithm uses compounds to deal with intermediate results it removes the load on the reduction and produces faster results. And, during the implementation, we must remember the type of slider input that should match the output type of connectors, which is exactly what our algorithm verifies.

TASK C.1

As the Y-crop company has a lot of data that is collected from drones, sensors, social media, online news feed so we can say that all of that data is dumped data, the data warehouse handles only the processed data so I believe that for this situation using the data lake gives us a larger capacity to store than the data warehouse this can help to quickly add data

This is very ideal for machine learning and when comes to Accessibility this can be very easily accessible when compared to the data warehouse, but it often feels very difficult to navigate the required data for unfamiliar people who are trying to use it. It offers unmatchable flexibility to the companies or domains to get clean insights about their data and it is relatively very cheap when compared to the other data warehouses a data lake can store multi-structured data from a number of resources in this case we are collecting from soil sensors, historical weather data, satellite images, drone videos of the crop and soil this will help how to maintain and monitor the quality of yield and whether the plants are getting cultivated properly or not.

As the volume of this company is expected to be more than 500 PB I prefer data lake is the better choice for creating good insights into the crop if it is a data warehouse the problem is the data that is coming from their sensors, drones, satellites are needed to be filtered before itself by the time it reaches to their database this will take a long process to upload the real-time data, in less time to import more resources and empowering users to share and analyse data in a variety of ways to make it better, faster decisions for the company.

“Data Lakes is an ideal operating load to be used in the cloud because the cloud provides performance, rating, reliability, availability, a diverse set of analytics engines, and a large scale economy. The ESG study found that 39% of respondents consider cloud as their primary analytical function, 41% for data storage, and 43% for Spark. The top reasons customers see the cloud as a benefit to Data Lakes is better security, faster feed time, better availability, feature/performance update, more flexibility, location coverage, and costs associated with actual use.” (Anon., n.d.)

TASK C.2

In this case, we need data storage to achieve lower delays and higher availability, the conditions of these applications need to be installed in nearby data centres and their users. I recommend Azure Cosmos DB because we need an application which can respond in real-time to large changes while using it, Azure Cosmos DB is fully controlled by NO SQL model.

Azure Cosmos DB removes website management from your hands with automatic management, updates, and amendments. It also handles power management with affordable server-free options and automated measurement options that respond to app requirements to match volume and demand.

The main challenge is as we have multi-un-structured data in this Y-crop company we are storing that data in a data lake first, so we need to write a python program to transfer the data in JSON format which stands for **J**ava**S**cript **O**bject **N**otation which is a text format for storing and transporting data, so if the data was successfully transfers from lake data (dump data)-> Azure Cosmos DB(JSON) then we can write queries very easily as this supports NO SQL this architecture helps us to take full advantage of the cloud to deliver zero downtime and this also use the clusters of computers which allows the data base to expand. So this will the company to write and build a data analytical store which can help to facilitate queries like

- “to find all the diseases which are caused by nitrogen deficiency ”
- “to find all the crops which are needed to be watered more”
- “We can also write queries like how much fertilizer was used in the last three years ”

This can support multiple data models using a single backend, it means it can be used for documentation, graph models, and key values as well. as the data is indexed automatically users can use access it using any API of their choice.

TASK C.3

No, I don't agree with the suggestion which was given by the IT managers because MapReduce cannot run real-time prediction because it needs to process batch-wise. In our case, the crop_y company deals with a lot of sensors, drones and satellite images these data might update within milliseconds. As the Hadoop MapReduce algorithm works very well with processed batch data so the problem arises when the data is updated in real-time. The alternative solution we can use is Kafka Streams this database will help to process the real-time prediction. Some of the events in the company are never-ending processes so it generates data very frequently for such kinds of situations we can use Kafka Streams database.

Using the MapReduce distributed processing framework will help the company to enhance the processing of the data by using scatted and parallel algorithms in the ecosystem. But it cannot handle real-time processing

In general, Hadoop was designed for batch processing so it executes jobs with a very fast run time. HDFS is generally designed by high throughput data I/O, rather than high performer I/O. The problem occurs when the main memory in a single system is processed on multiple servers.

Conclusion: some predictions and analytics provided by the Y-crop company require a response within a few seconds of the arrival of new data to handle this situation Hadoop MapReduce is not suitable rather we can Kafka Streams which will store the data by its time stamps after indicating the time duration the data will be get rid of the data, and this is an open-source platform as well we can use write the programs in java or Scala in this. So by considering all these factors we can say that this is best for real-time processing

MapReduce can be used to enhance the following factors but not the real time predictions and analytics tasks:

- This can help to increase the speed of the process of unstructured data
- It also helps the users to run their applications from many nodes
- It gives a very high scalable framework
- Allows the companies to store data in a cost-effective manner

TASK C.4

Design of cloud to increase the security, scalability & availability of the database

Security:

- Encryption is a method of protecting the data which is stored in the cloud it adds another layer of cloud security to protect your assets it is done by encoding the data with a key it's nearly impossible to decrypt without a key.
- Micro-segmentation is commonly used in cloud services it's a method of dividing your cloud into distinct security segments this will help the company to reduce and minimize the damage which can cause by a hacker who tries to gain the access to the data in the cloud.
- Firewalls this is a traditional method of securing the data this provides protection by including packet filtering, IP block, domain blocking, port blocking & proxying.
- Using (MFA) multi-factor authentication this will helps to stop the hackers if they stole the usernames and passwords, this MFA is commonly referred to as two-factor authentication for example this will send you (OTP) or push notification to the registered mobile device or by a third-party authentication.

Scalability:

- Load balancing on distributing computing network among serval hardware (i.e. drives, CPU, or separating in servers) the main goal of load balance is to boost network output and it even helps when a server crashes or fails, you will be automatically replaced with another.
- Auto-scaling will help by customizing the power of computing according to network load volume, this is a special method of approach to dynamic scaling. In recent times most companies such as AWS, google cloud, and Azure are using this method.

Availability:

- A good network connection is very important between the cloud and the local storage it needs a dedicated connectivity
- Monitoring can help to validate the uptime availability
- when an IP address fails it should be remapped to an alternative instance to redirect the traffic

Bibliography

Anon., n.d. *Amazon Web Services, Inc.*. [Online]
Available at: https://aws.amazon.com/big-data/datalakes-and-analytics/what-is-a-data-lake/

rimman (2019). Introduction to Azure Cosmos DB. [online] Microsoft.com. Available at: https://docs.microsoft.com/en-us/azure/cosmos-db/introduction.

TDAN.com. (n.d.). Big Data Hadoop vs. Traditional RDBMS. [online] Available at: https://tdan.com/big-data-hadoop-vs-traditional-rdbms [Accessed 25 Apr. 2022].