# LAB 10
# Big Data analysis with Hadoop: Extracting information from email records
### CO515: Advances in Computer Networks: Selected topics

## Objective

To provide hands-on experience with Hadoop for processing and extracting meaningful information from large email log datasets.

## Prerequisites:

⇒ Basic knowledge of Linux commands.
⇒ Familiarity with Java programming.
⇒ Understanding of big data concepts.
⇒ Basic understanding of email log formats.

## Activity 1: Setting up Hadoop Environment

**Install Java:**

- Ensure Java is installed on your machine. Hadoop requires Java to run.
  **sudo apt update**
  **sudo apt install openjdk-11-jdk**

**Download Hadoop:**

- Download the latest stable version of Hadoop from the official website.
  **wget https://downloads.apache.org/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz**

**Extract Hadoop:**

- Extract the downloaded Hadoop tar file.
  **tar -xzvf hadoop-3.3.0.tar.gz**

**Configure Hadoop:**

- Set up environment variables by adding the following lines to ~/.bashrc:
  **export HADOOP_HOME=/path/to/hadoop-3.3.0**
  **export PATH=$PATH:$HADOOP_HOME/bin**
  **export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64**
- Then, apply the changes:
  **source ~/.bashrc**

**Start Hadoop Services:**

- Format the HDFS and start Hadoop services.
  **hdfs namenode -format**
  **start-dfs.sh**

**start-yarn.sh**

## Activity 2: Loading Email Logs into HDFS

### Create Directories in HDFS:

- Create directories to store the input data.
  **hdfs dfs -mkdir -p /user/hadoop/input**

### Upload Data to HDFS:

- Upload the sample email log dataset to the HDFS input directory.
  **hdfs dfs -put /path/to/email-logs.csv /user/hadoop/input**

## Activity 3: Running a MapReduce Job to Extract Information

### Understand the Email Log Format:

- Sample email log entries might look like this:
  Ex: May 31 00:04:13 mail postfix/smtp[64637]: B91ED1F7DA: to=<namal@eng.pdn.ac.lk>, relay=relay.pdn.ac.lk[10.40.2.8]:25, delay=0.19, delays=0.12/0/0/0.07, dsn=2.0.0, status=sent (250 2.0.0 Ok: queued as 95F8B7865)

### Write a MapReduce Program:

- Create a Java MapReduce program to extract useful information, such as the count of emails received (to=<namal@eng.pdn.ac.lk ) by each sender.
- Shown below is a sample code.

```
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;

import java.io.IOException;

public class EmailLogAnalysis {

  public static class EmailMapper extends Mapper<Object, Text, Text, IntWritable>{
    private final static IntWritable one = new IntWritable(1);
    private Text sender = new Text();

    public void map(Object key, Text value, Context context) throws IOException,
InterruptedException {
      String[] fields = value.toString().split(" ");
      if(fields.length > 2) {
        sender.set(fields[3].replace("[", "").replace("]", ""));
        context.write(sender, one);
      }
    }
  }

  public static class EmailReducer 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 sum = 0;
      for (IntWritable val : values) {
        sum += val.get();
      }
      result.set(sum);
      context.write(key, result);
    }
  }

  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "email log analysis");
    job.setJarByClass(EmailLogAnalysis.class);
    job.setMapperClass(EmailMapper.class);
    job.setCombinerClass(EmailReducer.class);
    job.setReducerClass(EmailReducer.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);
  }
}
```

**Compile the Program:**

- Compile the Java program.

**javac -classpath `hadoop classpath` -d emaillog_classes EmailLogAnalysis.java**
**jar -cvf emaillog.jar -C emaillog_classes/ .**

**Run the MapReduce Job:**

- Run the compiled MapReduce job.

**hadoop jar emaillog.jar EmailLogAnalysis /user/hadoop/input /user/hadoop/output**

## Activity 4: Analyzing Results

**Check Job Output:**

- List the contents of the output directory.

**hdfs dfs -cat /user/hadoop/output/part-r-00000**

**View the Results:**

- View the contents of the output files.

**hdfs dfs -cat /user/hadoop/output/part-r-00000**

**Submission:** Submit the following documents to your instructor by the end of the lab session:

1. Code Implementation: Explanation of the MapReduce program written.
2. Results:Output from the MapReduce job. Any observations or insights from the email log analysis.