Hadoop

1) Выберите задачу, в алгоритме решения которой вы видите возможность эффективного распараллеливания. За основу рекомендуется взять табличные датасеты, преставленные в открытых источниках типа kaggle.com

https://www.kaggle.com/datasets/eliasdabbas/web-server-access-logs

`Поиск брутфорс-атак (много попыток с одного IP)`

2) Продумайте и опишите алгоритм функции Map и функции Reduce Реализуйте программный продукт, который будет реализовывать функцию Map, функцию Reduce, а также иные вспомогательные функции при их необходимости. Для этого выберите любую платформу и язык программирования, создайте в среде разработки проект, подключите необходимые библиотеки, напишите код и скомпилируйте его.

BruteforceMapper.java

```
package ru.muravev.mapreduce;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

import java.io.IOException;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class BruteforceMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
private static final String LOG_REGEX = "^([\\d.]+) (\\S+) (\\S+) \\[([^\\]]+)\\]
\"([^\"]+)\" (\\d{3}) (\\d+)";
private static final Pattern LOG_PATTERN = Pattern.compile(LOG_REGEX);

    private static final IntWritable ONE = new IntWritable(1);

    private Text ip = new Text();

    @Override
    protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text,
IntWritable>.Context context) throws IOException, InterruptedException {
        Matcher matcher = LOG_PATTERN.matcher(value.toString());
        if (matcher.find()) {
            ip.set(matcher.group(1));
            context.write(ip, ONE);
        }
    }
}
```

BruteforceReducer.java

```
package ru.muravev.mapreduce;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

import java.io.IOException;

public class BruteforceReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    private IntWritable result = new IntWritable();

    @Override
    protected void reduce(Text key, Iterable<IntWritable> values, Reducer<Text,
IntWritable, Text, IntWritable>.Context context) throws IOException,
InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}
```

BruteforceMain.java

```
package ru.muravev.mapreduce;

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.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;

public class BruteforceMain {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        String[] remainingArgs = new GenericOptionsParser(conf, args).
getRemainingArgs();

        if (remainingArgs.length != 2) {
            System.err.println("Must be 2 args: <input> <output>");
            System.exit(2);
        }
```

```
        Job job = Job.getInstance(conf, "Bruteforce");

        job.setJarByClass(BruteforceMain.class);
        job.setMapperClass(BruteforceMapper.class);
        job.setReducerClass(BruteforceReducer.class);
        job.setCombinerClass(BruteforceReducer.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        FileInputFormat.addInputPath(job, new Path(remainingArgs[0]));
        FileOutputFormat.setOutputPath(job, new Path(remainingArgs[1]));

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

```
mvn clean package
```

3) Перенесите разработанное приложение и его входные данные в кластер Hadoop и запустите приложение на выполнение. При этом входные данные должны быть такого размера, который бы позволил оценить целесообразность и преимущества обработки в распределенной системе.

Немного поменял docker-compose файл Hadoop'а

```
    volumes:
      - ./share:/share
```

Скопировал туда логи и jar'ник

```
hdfs dfs -mkdir -p /input
hdfs dfs -put /share/access_small.log /input/
hdfs dfs -ls /input
```

```
hadoop jar /share/bruteforce-detector-0.0.1.jar ru.muravev.mapreduce.BruteforceMain
/input/access_small.log /output
```

```
bash-4.2$ hadoop jar /share/bruteforce-detector-0.0.1.jar ru.muravev.mapreduce.BruteforceMain /input/access_small.log /output
2025-04-17 17:50:38 INFO  DefaultNoHARMFailoverProxyProvider:64 - Connecting to ResourceManager at resourcemanager/172.19.0.3:8032
2025-04-17 17:50:38 INFO  JobResourceUploader:907 - Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1744911347950_0002
2025-04-17 17:50:40 INFO  FileInputFormat:300 - Total input files to process : 1
2025-04-17 17:50:40 INFO  JobSubmitter:202 - number of splits:1
2025-04-17 17:50:40 INFO  JobSubmitter:298 - Submitting tokens for job: job_1744911347950_0002
2025-04-17 17:50:40 INFO  JobSubmitter:299 - Executing with tokens: []
2025-04-17 17:50:40 INFO  Configuration:2854 - resource-types.xml not found
2025-04-17 17:50:40 INFO  ResourceUtils:476 - Unable to find 'resource-types.xml'.
2025-04-17 17:50:40 INFO  YarnClientImpl:338 - Submitted application application_1744911347950_0002
2025-04-17 17:50:40 INFO  Job:1682 - The url to track the job: http://resourcemanager:8088/proxy/application_1744911347950_0002/
2025-04-17 17:50:40 INFO  Job:1727 - Running job: job_1744911347950_0002
2025-04-17 17:50:47 INFO  Job:1748 - Job job_1744911347950_0002 running in uber mode : false
2025-04-17 17:50:47 INFO  Job:1755 -  map 0% reduce 0%
2025-04-17 17:50:53 INFO  Job:1755 -  map 100% reduce 0%
2025-04-17 17:50:58 INFO  Job:1755 -  map 100% reduce 100%
2025-04-17 17:50:58 INFO  Job:1766 - Job job_1744911347950_0002 completed successfully
2025-04-17 17:50:58 INFO  Job:1773 - Counters: 54
        File System Counters
                FILE: Number of bytes read=1333
                FILE: Number of bytes written=555437
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
```

```
hdfs dfs -cat /output/part-r-00000
```

```
                Bytes Read=200480
        File Output Format Counters
                Bytes Written=1082
bash-4.2$ hdfs dfs -cat /output/part-r-00000
109.162.247.177 5
109.169.65.209  12
113.203.0.210   13
113.203.101.213 1
148.251.133.251 4
151.235.178.179 2
151.239.241.163 6
151.239.244.221 2
151.241.20.35   12
157.55.39.167   1
158.58.113.157  1
17.58.102.43    1
185.107.28.2    13
185.161.113.50  1
188.158.191.102 28
188.159.73.223  13
188.34.54.0     1
192.15.168.184  2
192.15.67.203   1
195.181.168.181 8
```

part-r-00000

```
109.162.247.177 5
109.169.65.209  12
113.203.0.210   13
113.203.101.213 1
148.251.133.251 4
151.235.178.179 2
151.239.241.163 6
151.239.244.221 2
151.241.20.35   12
```

```
157.55.39.167   1
158.58.113.157  1
17.58.102.43    1
185.107.28.2    13
185.161.113.50  1
188.158.191.102 28
188.159.73.223  13
188.34.54.0     1
192.15.168.184  2
192.15.67.203   1
195.181.168.181 8
2.178.172.239   11
2.179.13.33     20
204.18.253.65   8
40.77.167.103   1
46.209.207.227  22
46.224.62.57    4
46.32.7.230     13
5.106.130.52    11
5.112.240.241   1
5.112.94.17     1
5.114.86.57     37
5.115.243.31    1
5.116.118.58    10
5.117.210.134   27
5.120.174.159   3
5.120.36.176    1
5.122.25.167    17
5.125.149.186   1
5.134.145.80    2
5.208.12.192    7
5.208.194.243   2
5.209.8.169     1
5.210.86.107    28
5.211.9.217     1
5.52.244.220    20
5.74.173.136    1
63.143.42.246   1
65.49.68.185    2
66.249.66.194   17
66.249.66.91    6
66.249.66.92    1
78.39.200.178   47
80.250.199.165  6
82.99.235.200   6
83.120.50.196   4
83.121.228.101  1
83.121.95.172   1
83.122.164.249  1
89.221.88.241   5
89.38.197.213   16
```

```
91.99.30.32     6
92.50.40.46     1
93.118.108.45   1
95.216.86.214   22
95.64.78.241    1
95.80.164.20    17
95.85.48.18     24
....
```

5) При помощи встроенных средств Hadoop продемонстрируйте распределение вычислений по узлам кластера.

```
bash-4.2$ yarn node -list
2025-04-17 18:23:54 INFO  DefaultNoHARMFailoverProxyProvider:64 - Connecting to ResourceManager at resourcemanager/172.19.0.2:8032
Total Nodes:4
         Node-Id             Node-State Node-Http-Address       Number-of-Running-Containers
2d9d445341fd:34635              RUNNING 2d9d445341fd:8042                                  4
5befa8db5ebd:40025              RUNNING 5befa8db5ebd:8042                                  0
5befa8db5ebd:36105              RUNNING 5befa8db5ebd:8042                                  5
dab218eed590:39541              RUNNING dab218eed590:8042                                  3
bash-4.2$
```

6) Продемонстрируйте результат обработки данных. Например, откройте сгенерированный выходной файл с результатами.