

assignment1

October 6, 2024

1 Assignment 1

1.1 Question 1

Consider a Map/Reduce solution to a program that reads a file full with integers and computes the number of integers that are divisible by 3. Each line can contain multiple integers. Example input:

2 324 234 234 234

21432 324 3243 2143214

3214324 24 223 454

- a. What is the input/output type of the mapper and reducer?
 - Mapper: (LongWritable, Text) -> (Text, IntWritable)
 - Reducer: (Text, Iterable<IntWritable>) -> (Text, IntWritable)
- b. Show the map method for the program.

```
import java.io.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.Mapper.*;

public class MyMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
    @Override
    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
        String line = value.toString();
        String[] numbers = line.split(" ");
        for (String number : numbers) {
            int num = Integer.parseInt(number);
            if (num % 3 == 0) {
                context.write(new Text("count"), new IntWritable(1));
            }
        }
    }
}
```

- c. Show the reduce method for the program.

```
import java.io.*;
import org.apache.hadoop.io.*;
```

```

import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.Reducer.*;

public class MyReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    @Override
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException {
        int count = 0;
        for (IntWritable value : values) {
            count += value.get();
        }
        context.write(key, new IntWritable(count));
    }
}

```

- d. Can you create a combiner for the program? What will be the code for the reduce method of the combiner? Do you need to change the map or reduce method to support the combiner? If so, show the new map and reduce method.

- Yes, we can create a combiner for the program:

```

public class MyCombiner extends Reducer<Text, IntWritable, Text, IntWritable> {
    @Override
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException {
        int count = 0;
        for (IntWritable value : values) {
            count += value.get();
        }
        context.write(key, new IntWritable(count));
    }
}

```

- We don't need to change the map or reduce method to support the combiner.

1.2 Question 2

Consider a Map/Reduce solution to a program that reads a file that contains the date and a temperature that is measured for this day on each line. There can be multiple temperatures that are measured each day. The program should calculate the highest temperature for each day. Example input:

2022/12/10 10

2022/12/10 8

2022/12/10 6

2022/12/12 11

2000/01/01 3

- What is the input/output type of the mapper and reducer?
 - Mapper: (LongWritable, Text) -> (Text, IntWritable)

- Reducer: (Text, Iterable<IntWritable>) -> (Text, IntWritable)

b. Show the map method for the program.

```
import java.io.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.Mapper.*;

public class MyMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
    @Override
    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
        String line = value.toString();
        String[] parts = line.split(" ");
        if (parts.length != 2) {
            return;
        }
        String date = parts[0];
        int temperature = Integer.parseInt(parts[1]);
        context.write(new Text(date), new IntWritable(temperature));
    }
}
```

c. Show the reduce method for the program.

```
import java.io.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.Reducer.*;

public class MyReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    @Override
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {
        int maxTemperature = Integer.MIN_VALUE;
        for (IntWritable value : values) {
            maxTemperature = Math.max(maxTemperature, value.get());
        }
        context.write(key, new IntWritable(maxTemperature));
    }
}
```

d. Can you create a combiner for the program? What will be the code for the reduce method of the combiner? Do you need to change the map or reduce method to support the combiner? If so, show the new map and reduce method.

- Yes, we can create a combiner for the program:

```
public class MyCombiner extends Reducer<Text, IntWritable, Text, IntWritable> {
    @Override
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {
        int maxTemperature = Integer.MIN_VALUE;
```

```
        for (IntWritable value : values) {
            maxTemperature = Math.max(maxTemperature, value.get());
        }
        context.write(key, new IntWritable(maxTemperature));
    }
}
```

- We don't need to change the map or reduce method to support the combiner, since the reducer still needs to handle multiple max values from different mappers.