CS5425 Assignment2 Task1 Report

Name:Guo Shijia   ID:A0191309E

Implementation detail:

1) It is quite simple to implement this by using spark, it have a lot of API to use:
Something need to specify is how we clean the token: we use regex to extract the words:

```
def clean(word: String): String={
  try {
        val pattern: Regex = "([A-Za-z]+)".r
        val pattern(res) = word
        return res.toLowerCase()
    } catch {
        case e: Exception => return ""
    }
  }
```

(We also filter out the words that length less than 2)

2)How we compute the common words in 2 RDDs(pairs(word, frequency))

```
val joinedRDD = input1_counts.join(input2_counts)
val result = joinedRDD.mapValues(x => List(x._1,
x._2).min).sortBy(_._2,false).take(15)
```

By using join, we filter the words that not common, we get pairs(words, [frequency1,frequency2]), The mapValues operation to keep the low frequency.

3)The other API We use is quite common, like map, flatMap, filter, reduceByKey, join, mapValues, sortBy, etc. The implementation is quite short, the whole code is about 40 lines(including 2 help functions), you can refer detail in source code.

Comparisons on programming with Hadoop and Spark

The most difference is that programming in Spark is more simple and more elegant than Hadoop. Spark have a lot of build-in API, which can help us to write less code.

Pros and Cons (Hadoop)

The framework is more clear, we can split the work into different map-reduce phases. There has more space leave to the developer, which need strong coding skills to implement difficult tasks. It

also need write lengthy code to implement a simple task. It will produce a lot of inner files which it's increase the cost and reduce the efficiency, but make the work is more fault-tolerant.

Pros and Cons (Spark)

The API is powerful and you can write less code. The whole process flow is in the memory, which need more memory in Spark env. Because the computation is based on memory, it's more volatile and more risky.

Comparisons on execution time with Hadoop and Spark

we run the same job in single node in both spark and hadoop, the average execution time is:

Spark: 4577 ms   Hadoop: 6120 ms

We can see the different execution time between Hadoop and Spark, from the result we can observe, the Spark is more efficient. (There maybe error brings in the implementation, we include the save result time and have different clean words implementation)

Pros and Cons (Hadoop)

Hadoop may spend more time than Spark to execute the same job, because it write the inner result into HDFS. It will cost more time but it’s more safe. If in some cases, the power is off, when the machine is up , we can continue our job from the inner result.

Pros and Cons (Spark)

Spark may spend less time than Hadoop, because it’s do not need to store the inner result into file system. But it will consume more memory than Hadoop and will lost all information when power is off.