master 터미널

# start-dfs.sh

# start-yarn.sh

# jps

> 6개 다 뜨는거 확인

slave 터미널

# jps

> 3개 뜨는거 확인

http://192.168.10.1:50070 띄워놓기

Mapper

> 사용자로부터 제일 먼저 데이터를 받는곳

> key와 value를 하낙로 묶는 역할

> Combiner로 네용 전달

Mapper -> Combiner -> Reducer

MyMapper.java

==============

package count;

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

// 1, read a book

// 2, writ a book

// <read, 1> <a, 1> <book, 1>

public class MyMapper extends Mapper<LongWritable, Text, Text, IntWritable>{

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

private Text word = new Text();

@Override

protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text, IntWritable>.Context context)

throws IOException, InterruptedException {

StringTokenizer token = new StringTokenizer(value.toString());

// 문장 갈라주기

while(token.hasMoreTokens()) {

word.set(token.nextToken());

context.write(word, one);

}

}

}

------------------------------------------

Source > Override

> map(LongWritable, Text, Context) 체크

============================================

MyReducer.java

===============

Source > Override

> reducer 체크

**package** count;

**import** java.io.IOException;

**import** org.apache.hadoop.io.IntWritable;

**import** org.apache.hadoop.io.Text;

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

**public** **class** MyReducer **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 ctx) **throws** IOException, InterruptedException {

**int** sum = 0;

**for**(IntWritable val : values) {

sum += val.get();

}

result.set(sum);

ctx.write(key, result);

}

}

============================================

WordCount.java

===============

**package** count;

**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.lib.input.FileInputFormat;

**import** org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

**import** org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

**import** org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

**public** **class** WordCount {

**public** **static** **void** main(String[] args) **throws** IOException, ClassNotFoundException, InterruptedException {

Configuration conf = **new** Configuration();

**if**(args.length != 2) {

System.***out***.println("사용방법 : WordCount <input> <output>");

System.*exit*(1); //강제 종료

}

Job job = Job.*getInstance*(conf, "WordCount");

job.setJarByClass(WordCount.**class**);

job.setMapperClass(MyMapper.**class**);

job.setReducerClass(MyReducer.**class**);

job.setInputFormatClass(TextInputFormat.**class**);

job.setOutputFormatClass(TextOutputFormat.**class**);

job.setOutputKeyClass(Text.**class**);

job.setMapOutputValueClass(IntWritable.**class**);

FileInputFormat.*addInputPath*(job, **new** Path(args[0]));

FileOutputFormat.*setOutputPath*(job, **new** Path(args[1]));

job.waitForCompletion(**true**);

}

}

Hadoop >오른쪽 마우스 > export >JARFile>Overwrite에 체크

winscp에서 export한 Hadoop.jar 파일을 리눅스 source 폴더에 덮어쓰기 하기

마스터에서

hdfs dfs -ls /upload // input.txt있는지 확인

cd source

hadoop jar Hadoop.jar count.WordCount /upload/input.txt ~/wordcount\_output1

hdfs dfs -ls ~/wordcount\_output1

hdfs dfs -cat ~/wordcount\_output1/part-r-00000 //글자수 세기 잘 나오는 지 확인

/////////// /자바프로그래밍 3 ////////////////////////

이클립스

Hadoop에 패키지 추가 name : sort

sort 패키지 내부에 class 추가 StringSort , public static 체크

**package** sort;

**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.KeyValueTextInputFormat;

**import** org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat;

**import** org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

**import** org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

**import** org.apache.hadoop.mapreduce.lib.output.SequenceFileOutputFormat;

**import** org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

**public** **class** StringSort {

**public** **static** **void** main(String[] args) **throws** Exception {

Configuration conf = **new** Configuration();

**if**(args.length != 2) {

System.***out***.println("사용방법 : StringSort <input> <output>");

System.*exit*(1); //강제 종료

}

Job job = Job.*getInstance*(conf, "StringSort");

job.setJarByClass(StringSort.**class**);

job.setMapperClass(Mapper.**class**);

job.setReducerClass(Reducer.**class**);

job.setInputFormatClass(KeyValueTextInputFormat.**class**);

job.setOutputFormatClass(SequenceFileOutputFormat.**class**);

job.setOutputKeyClass(Text.**class**);

job.setOutputValueClass(Text.**class**);

job.setMapOutputKeyClass(Text.**class**);

job.setMapOutputValueClass(Text.**class**);

job.setNumReduceTasks(1);

SequenceFileInputFormat.*addInputPath*(job, **new** Path(args[0]));

SequenceFileOutputFormat.*setOutputPath*(job, **new** Path(args[1]));

job.waitForCompletion(**true**);

}

}

Hadoop >오른쪽 마우스 > export >JARFile>Overwrite에 체크

winscp에서 export한 Hadoop.jar 파일을 리눅스 source 폴더에 덮어쓰기 하기

시놀에서 다운받은 randomstring.txt 도 옮기기

cd /root/source

hdfs dfs -put RandomString.txt /upload/ RandomString.txt

hdfs dfs -ls ~/random\_output

hdfs dfs -text ~/random\_output/part-r-00000 | head -200 //cat :텍스트만 볼수있다. 따라서 text 명령어 써서 바이너리 파일도 볼 수 있도록.

이클립스 코드

**package** sort;

**import** org.apache.hadoop.conf.Configuration;

**import** org.apache.hadoop.fs.Path;

**import** org.apache.hadoop.io.IntWritable;

**import** org.apache.hadoop.io.SequenceFile;

**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.KeyValueTextInputFormat;

**import** org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat;

**import** org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

**import** org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

**import** org.apache.hadoop.mapreduce.lib.output.SequenceFileOutputFormat;

**import** org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

**public** **class** StringSort {

**public** **static** **void** main(String[] args) **throws** Exception {

Configuration conf = **new** Configuration();

**if**(args.length != 2) {

System.***out***.println("사용방법 : StringSort <input> <output>");

System.*exit*(1); //강제 종료

}

Job job = Job.*getInstance*(conf, "StringSort");

job.setJarByClass(StringSort.**class**);

job.setMapperClass(Mapper.**class**);

job.setReducerClass(Reducer.**class**);

job.setInputFormatClass(KeyValueTextInputFormat.**class**);

job.setOutputFormatClass(SequenceFileOutputFormat.**class**);

job.setOutputKeyClass(Text.**class**);

job.setOutputValueClass(Text.**class**);

job.setMapOutputKeyClass(Text.**class**);

job.setMapOutputValueClass(Text.**class**);

job.setNumReduceTasks(1);

SequenceFileInputFormat.*addInputPath*(job, **new** Path(args[0]));

SequenceFileOutputFormat.*setOutputPath*(job, **new** Path(args[1]));

SequenceFileOutputFormat.*setOutputCompressionType*(job, SequenceFile.CompressionType.***BLOCK***);

job.waitForCompletion(**true**);

}

}

수정 (stringsort) 또 export, 리눅스로 옮기기

hdfs dfs -rm -r -f ~/random\_output

hadoop jar Hadoop.jar sort.StringSort /upload/RandomString.txt ~/random-output