

File Machine View Input Devices Help

Applications Places System



File Edit View Search Terminal Help

```
Map output records=10
Map output bytes=110
Map output materialized bytes=69
Input split bytes=124
Combine input records=10
Combine output records=5
Reduce input groups=5
Reduce shuffle bytes=69
Reduce input records=5
Reduce output records=5
Spilled Records=10
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=175
CPU time spent (ms)=2830
Physical memory (bytes) snapshot=347697152
Virtual memory (bytes) snapshot=3007156224
Total committed heap usage (bytes)=226365440
```

#### Shuffle Errors

```
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
```

#### File Input Format Counters

```
Bytes Read=70
```

#### File Output Format Counters

```
Bytes Written=43
```

```
[cloudera@quickstart ~]$ hdfs dfs -ls /out1
```

```
Found 2 items
```

```
-rw-r--r-- 1 cloudera supergroup 0 2025-02-04 03:13 /out1/_SUCCESS
-rw-r--r-- 1 cloudera supergroup 43 2025-02-04 03:13 /out1/part-r-00000
```

```
[cloudera@quickstart ~]$ hdfs dfs -cat /out1/part-r-00000
```

```
Bramhecha 3
```

```
DSBDA 2
```

```
PICT 2
```

```
Parth 2
```

```
parth 1
```

```
[cloudera@quickstart ~]$ cat /home/cloudera/Procesfile.txt
```

```
Parth Bramhecha
```

```
parth Bramhecha
```

```
PICT
```

```
PICT
```

```
DSBDA
```

```
Parth Bramhecha
```

```
DSBDA
```

cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Applications Places System

Tue Feb 4, 3:19 AM cloudera

cloudera@quickstart:~

```
[cloudera@quickstart ~]$ ls
cloudera-manager  eclipse          Music      WordCount.jar
cm_api.py         enterprise-deployment.json  Pictures   workspace
Desktop           express-deployment.json    Public     workspace1
Documents          kerberos          Templates
Downloads          lib                Videos

[cloudera@quickstart ~]$ pwd
/home/cloudera
[cloudera@quickstart ~]$ cat > /home/cloudera/Procesfile.txt
Parth Bramhecha
parth Bramhecha
PICT
PICT
DSBDA
Parth Bramhecha
DSBDA
^C
[cloudera@quickstart ~]$ cat /home/cloudera/Procesfile.txt
Parth Bramhecha
parth Bramhecha
PICT
PICT
DSBDA
Parth Bramhecha
DSBDA
[cloudera@quickstart ~]$ hdfs dfs -ls
Found 1 items
drwxr-xr-x  - cloudera cloudera      0 2025-01-21 09:25 wordcount
[cloudera@quickstart ~]$ hdfs dfs -ls/
-ls/: Unknown command
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 5 items
drwxr-xr-x  - hbase supergroup      0 2025-02-04 02:52 /hbase
drwxr-xr-x  - solr  solr          0 2015-06-09 03:38 /solr
drwxrwxrwx  - hdfs supergroup      0 2025-01-21 09:20 /tmp
drwxr-xr-x  - hdfs supergroup      0 2015-06-09 03:38 /user
drwxr-xr-x  - hdfs supergroup      0 2015-06-09 03:36 /var
[cloudera@quickstart ~]$ hdfs dfs -mkdir /inputfolder1
[cloudera@quickstart ~]$ hdfs dfs -put /home/cloudera/Processfile.txt /inputfolder1/
put: `/home/cloudera/Processfile.txt': No such file or directory
[cloudera@quickstart ~]$ pwd
/home/cloudera
[cloudera@quickstart ~]$ ls
cloudera-manager  Desktop        Downloads  enterprise-deployment.json  kerberos  Music      Procesfile.txt  Templates  WordCount.jar  workspace1
cm_api.py        Documents       eclipse    express-deployment.json    lib      Pictures   Public      Videos   workspace
[cloudera@quickstart ~]$ hdfs dfs -put /home/cloudera/Procesfile.txt /inputfolder1/
put: `/home/cloudera/Procesfile.txt': No such file or directory
[cloudera@quickstart ~]$ hdfs dfs -put /home/cloudera/Procesfile.txt /inputfolder1/
[cloudera@quickstart ~]$ hdfs dfs -cat /inputfolder1/Procesfile.txt
```

## WordCount.java

```
1 import java.io.IOException;
2 import java.util.StringTokenizer;
3
4 import org.apache.hadoop.conf.Configuration;
5 import org.apache.hadoop.fs.Path;
6 import org.apache.hadoop.io.IntWritable;
7 import org.apache.hadoop.io.Text;
8 import org.apache.hadoop.mapreduce.Job;
9 import org.apache.hadoop.mapreduce.Mapper;
10 import org.apache.hadoop.mapreduce.Reducer;
11 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
12 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
13
14 public class WordCount {
15
16     public static class TokenizerMapper
17         extends Mapper<Object, Text, Text, IntWritable>{
18
19         private final static IntWritable one = new IntWritable(1);
20         private Text word = new Text();
21
22         public void map(Object key, Text value, Context context
23                         ) throws IOException, InterruptedException {
24             StringTokenizer itr = new StringTokenizer(value.toString());
25             while (itr.hasMoreTokens()) {
26                 word.set(itr.nextToken());
27                 context.write(word, one);
28             }
29         }
30     }
31
32     public static class IntSumReducer
33         extends Reducer<Text,IntWritable,Text,IntWritable> {
34         private IntWritable result = new IntWritable();
35
36         public void reduce(Text key, Iterable<IntWritable> values,
37                           Context context
38                           ) throws IOException, InterruptedException {
39             int sum = 0;
40             for (IntWritable val : values) {
41                 sum += val.get();
```

```
public void map(Object key, Text value, Context context
                ) throws IOException, InterruptedException {
    StringTokenizer itr = new StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
    }
}

public static class IntSumReducer
    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, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.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);
}
```