

CS236 Final Project

MapReduce

Mapper

- Input: key/value pairs
- Output: sorted set of intermediate key/value pairs
- The intermediate results are partitioned per Reducer

Example Mapper Code

```
public void map(LongWritable key, Text value,
OutputCollector<Text, IntWritable> output, Reporter
reporter) throws IOException {
    String line = value.toString();
    StringTokenizer tokenizer = new
StringTokenizer(line);
    while (tokenizer.hasMoreTokens()) {
        word.set(tokenizer.nextToken());
        output.collect(word, one);
    }
}
```

WordCount Example

Input File
1:

Hello World
Bye World

Intermediate
Result:

<Bye, 1>
<Hello, 1>
<World, 1>
<World, 1>

Input File
2:

Hello Hadoop
Bye Hadoop

Intermediate
Result:

<Bye, 1>
<Hadoop, 1>
<Hadoop, 1>
<Hello, 1>

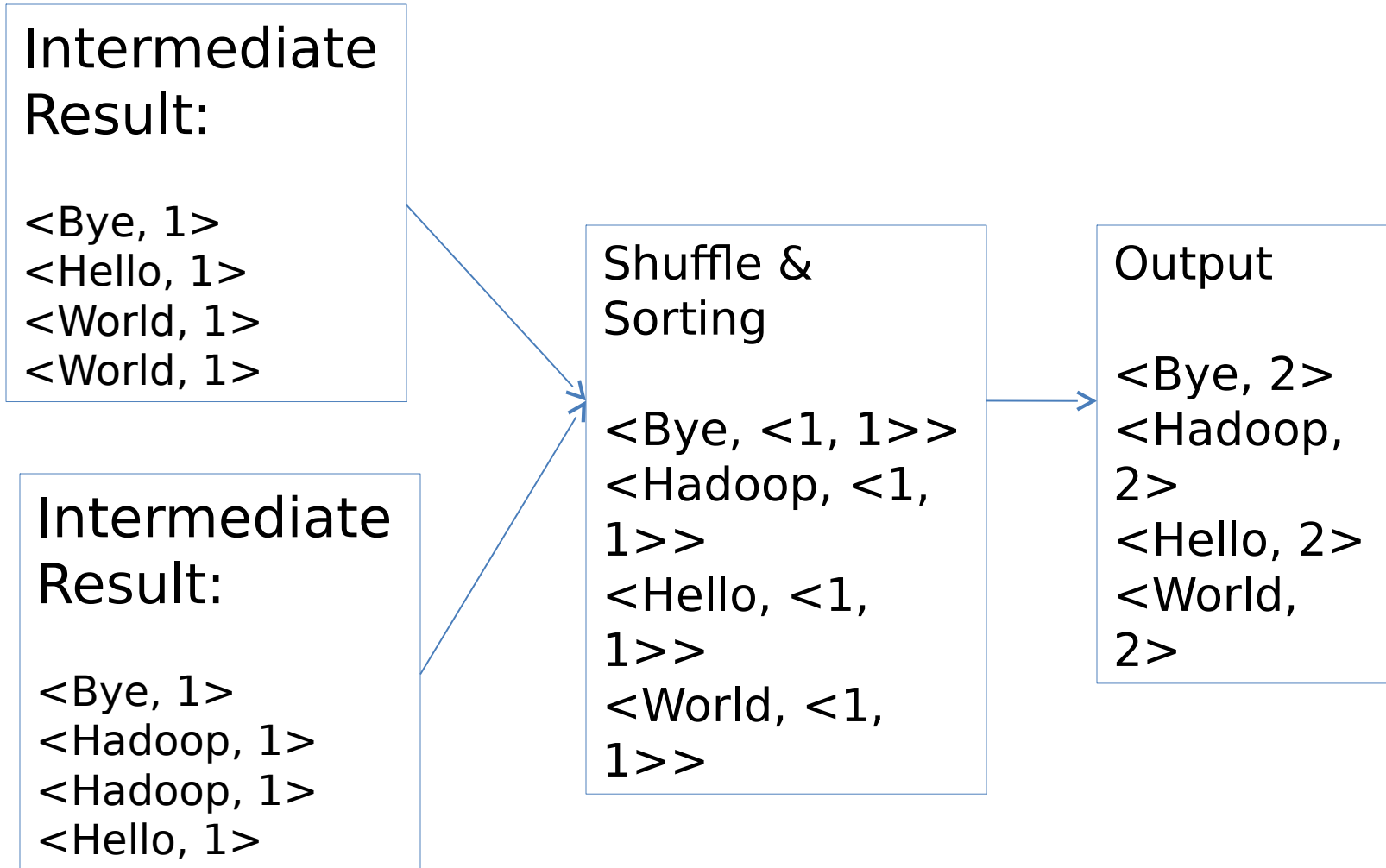
Reducer

- Input: mapper output
- Shuffling and Sorting
- Output: Reduced key/value pairs

Example Reducer Code

```
public void reduce(Text key, Iterator<IntWritable>
values, OutputCollector<Text, IntWritable> output,
Reporter reporter) throws IOException {
    int sum = 0;
    while (values.hasNext()) {
        sum += values.next().get();
    }
    output.collect(key, new IntWritable(sum));
}
```

WordCount Example - Cont.



Main Program

```
public static void main(String[] args) throws Exception {
    JobConf conf = new JobConf(WordCount.class);
    conf.setJobName("wordcount");

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

    conf.setMapperClass(Map.class);
    conf.setReducerClass(Reduce.class);

    conf.setInputFormat(TextInputFormat.class);
    conf.setOutputFormat(TextOutputFormat.class);

    FileInputFormat.setInputPaths(conf, new Path(args[0]));
    FileOutputFormat.setOutputPath(conf, new Path(args[1]));

    JobClient.runJob(conf);
}
```