Hadoop Streaming

Wordcount Demo

Outline

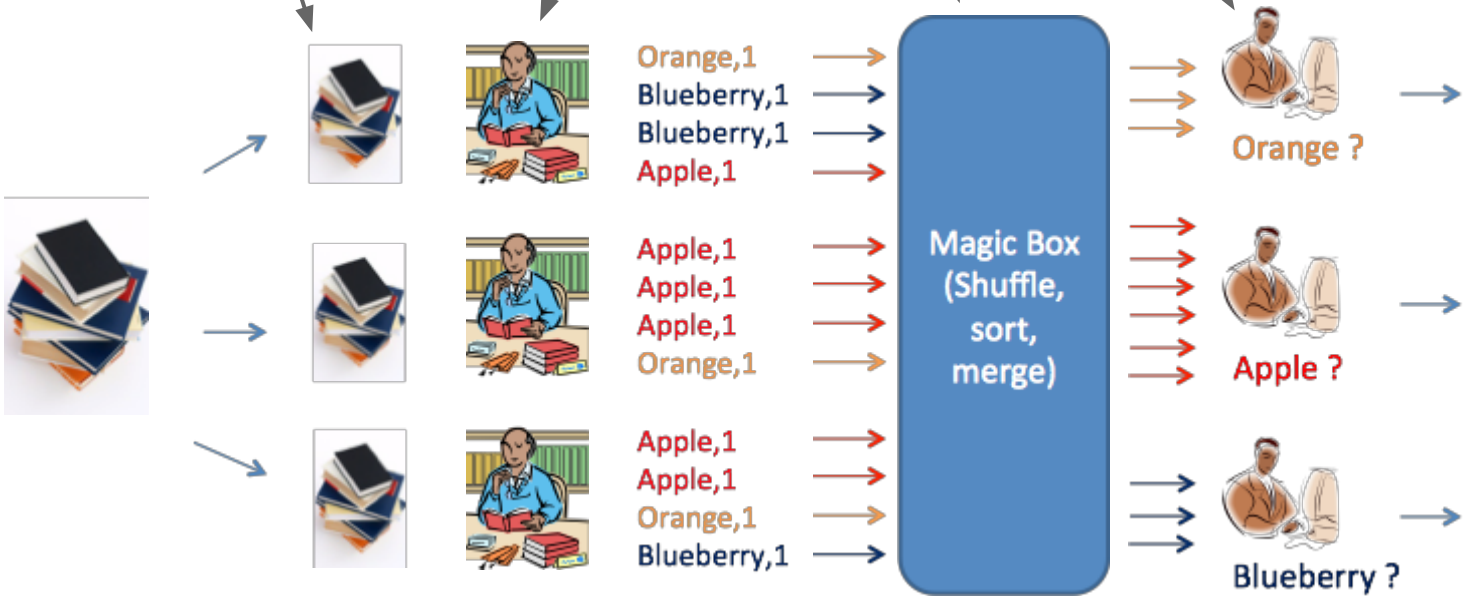
- Demo1:
  - Introduction to Hadoop streaming
  - Sample code for wordcount
- Demo2: s3 operations
- Demo3:
  - Run wordcount with Amazon EMR console

Hadoop Streaming

- Represent streaming via unix pipes (locally)

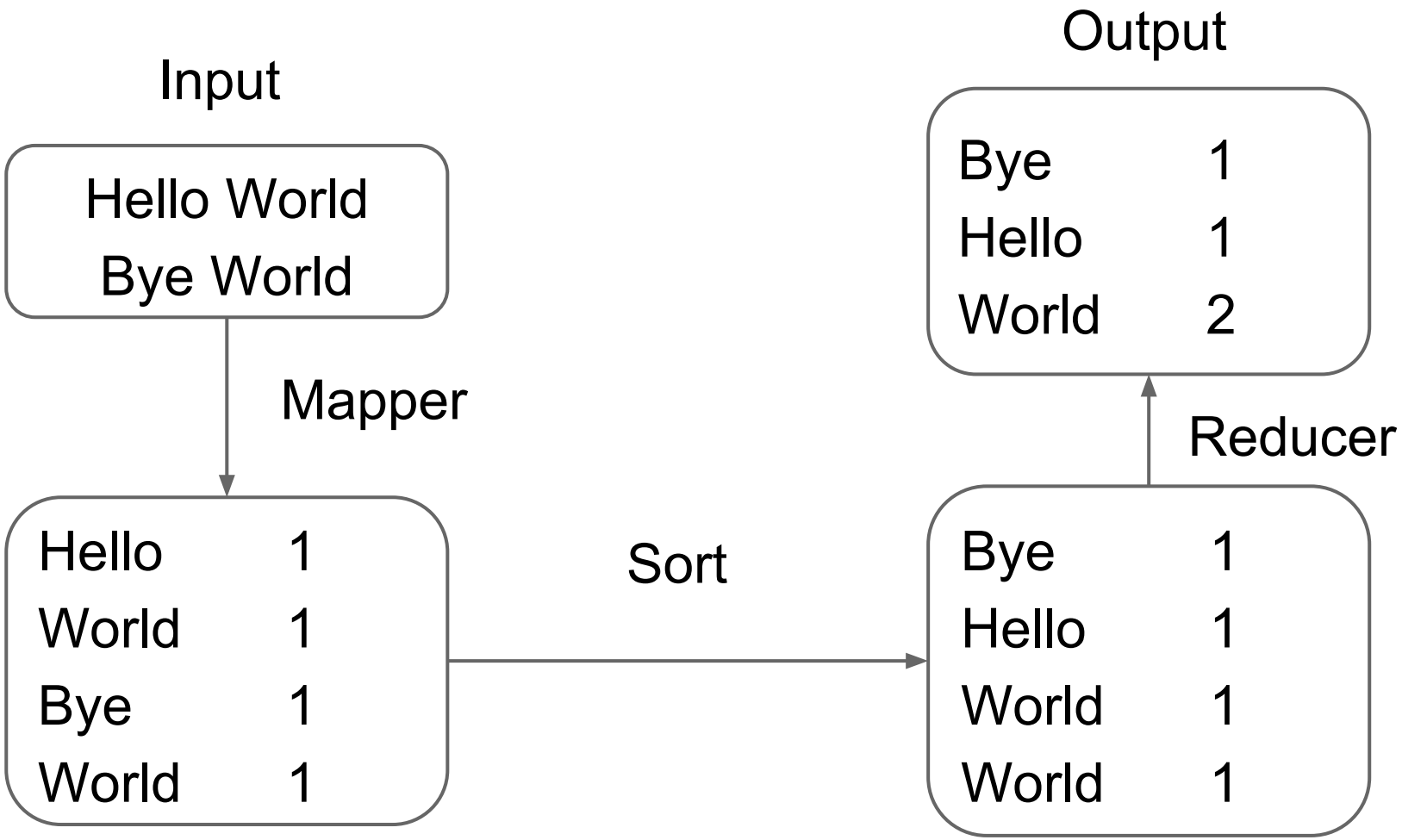
```
cat input.txt | mapper.py | sort | reducer.py
```

- How does it work in Mapreduce?



WordCount

WordCount


Input
Hello World
Bye World
Mapper
Hello 1
World 1
Bye 1
World 1
Sort
Bye 1
Hello 1
World 1
World 1
Reducer
Output
Bye 1
Hello 1
World 2

WordCount: wc-mapper.py

s3://wc-demo/wc-mapper.py

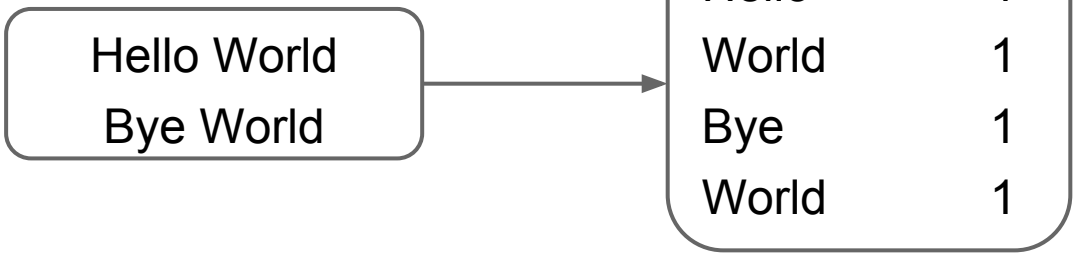
```
1.  #!/usr/bin/python
2.
3.  import sys
4.  import re
5.
6.  def main(argv):
7.      pattern = re.compile("[a-zA-Z][a-zA-Z0-9]*")
8.      for line in sys.stdin:
9.          for word in pattern.findall(line):
10.             print word.lower() + "\t" + "1"
11.
12. if __name__ == "__main__":
13.     main(sys.argv)
```

Remember to add this!!

1. Read line by line

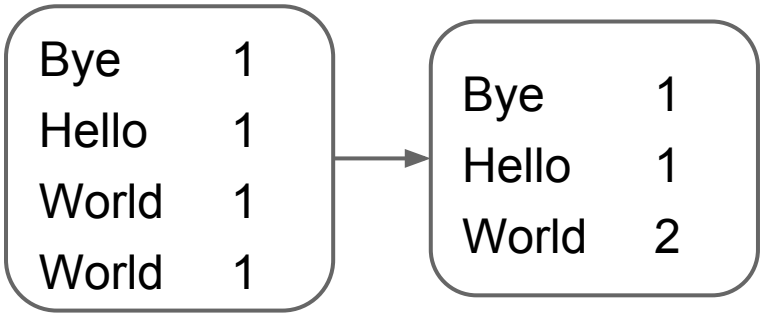
2. Tokenize

3. Generate (key, value) pair



WordCount: wc-reducer.py

s3://wc-demo/wc-reducer.py




```
 6.  (last_word, sum) = (None, 0)
 7.
 8.  # input comes from STDIN
 9.  for line in sys.stdin:
10.      # parse the input we got from mapper.py
11.      (cur_word, value) = line.strip().split('\t')
12.
13.      if last_word and last_word != cur_word:
14.          # write result to STDOUT
15.          print last_word + '\t' + str(sum)
16.          (last_word, sum) = (cur_word, int(value))
17.      else:
18.          (last_word, sum) = (cur_word, sum + int
    (value))
19.
20.  if last_word:
21.      print last_word + '\t' + str(sum)
```

Resources

- Python Tutorial http://docs.python.org/2/tutorial/
- Elastic MapReduce http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-what-is-emr.html
- Spot instance:
  - http://aws.amazon.com/ec2/spot-instances/