MapReduce (M/R)

Part 2

What Have We Done in Part 1?

- Wrote a map function and a reduce function for analyzing the NCDC global temperature dataset.
- The M/R functions and the STDIN/STDOUT were chained together forming a pipeline.
- The maximum temperatures for 1901 and 1902 were correctly pulled out from the dataset.
- How much was the Hadoop platform involved in above procedure?
  A. Hadoop was involved in all the steps.
  B. Only in the stage of the map function.
  C. Only in the stage of the reduce function.
  D. Not involved at all.

Hadoop Streaming

- A Hadoop API to MapReduce that allows writing the map and reduce functions in languages other than Java.
- It uses Linux/Unix standard streams as the interface between Hadoop and your program, so you can use any programming language that can read standard input and write to standard output to write your MapReduce programs.
- A map output key/value pair is written as a single tab-delimited line. Input to the reduce function is in the same format, a tab-separated key/value pair, passed over standard input. The reduce function reads lines from standard input, which the framework guarantees to be sorted by keys, and writes its results to standard output.

What to Do?

- Use the Hadoop Streaming to pass data from the Mapper to the Reducer via STDIN (standard input) and STDOUT (standard output).
- Just simply use Python's sys.stdin to read input data and print the output to sys.stdout.
- The rest is taken care by Hadoop Streaming API!

Use Hadoop Streaming to Process the Temperature Data (for HDP 2.5)

- Put your executable mapper and reducer Python programs in current directory; otherwise, give the full path of these files.
- Run Python M/R functions against the 1900-1 global temperature dataset through Hadoop streaming with the following commands at the CentOS prompt:

```
hadoop jar /usr/hdp/2.5.0.0-1245/ \
hadoop-mapreduce/hadoop-streaming.jar  \
-file temp_map.py \
-file temp_reduce.py \
-mapper temp_map.py \
-reducer temp_reduce.py \
-input /stsci5065/data/1901-1\
-output /stsci5065/maxtemp-output
```

Use Hadoop Streaming to Process the Temperature Data (for HDP 2.6.1)

- Put your executable mapper and reducer Python programs in current directory; otherwise, give the full path of these files.
- Run Python M/R functions against the 1901-1 global temperature dataset through Hadoop streaming with the following commands at the CentOS prompt:

```
hadoop jar /usr/hdp/2.6.1.0-129/ \
hadoop-mapreduce/hadoop-streaming.jar  \
-file temp_map.py \
-file temp_reduce.py \
-mapper temp_map.py \
-reducer temp_reduce.py \
-input /stsci5065/data/1901-1\
-output /stsci5065/maxtemp-output
```

The Hadoop Streaming Was Successful: The Beginning of the Processing Log

```
[root@sandbox stsci5065L]# hadoop jar /usr/hdp/2.5.0.0-1245/hadoop-mapreduce/hadoop-streamin
g.jar  -file temp_map.py  -file temp_reduce.py  -mapper temp_map.py  -reducer temp_reduce.py
  -input /stsci5065/data/1901-1 -output /stsci5065/maxtemp-output1
19/02/21 16:06:58 WARN streaming.StreamJob: -file option is deprecated, please use generic o
ption -files instead.
packageJobJar: [temp_map.py, temp_reduce.py] [/usr/hdp/2.6.1.0-129/hadoop-mapreduce/hadoop-s
treaming-2.7.3.2.6.1.0-129.jar] /tmp/streamjob3113860336361487675.jar tmpDir=null
19/02/21 16:06:58 INFO client.RMProxy: Connecting to ResourceManager at sandbox.hortonworks.
com/172.17.0.2:8032
19/02/21 16:06:59 INFO client.AHSProxy: Connecting to Application History server at sandbox.
hortonworks.com/172.17.0.2:10200
19/02/21 16:06:59 INFO client.RMProxy: Connecting to ResourceManager at sandbox.hortonworks.
com/172.17.0.2:8032
19/02/21 16:06:59 INFO client.AHSProxy: Connecting to Application History server at sandbox.
hortonworks.com/172.17.0.2:10200
19/02/21 16:06:59 INFO mapred.FileInputFormat: Total input paths to process : 1
19/02/21 16:07:00 INFO mapreduce.JobSubmitter: number of splits:2
19/02/21 16:07:00 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1550718539510_
0009
19/02/21 16:07:00 INFO impl.YarnClientImpl: Submitted application application_1550718539510_
0009
19/02/21 16:07:00 INFO mapreduce.Job: The url to track the job: http://sandbox.hortonworks.c
om:8088/proxy/application_1550718539510_0009/
19/02/21 16:07:00 INFO mapreduce.Job: Running job: job_1550718539510_0009
19/02/21 16:07:06 INFO mapreduce.Job: Job job_1550718539510_0009 running in uber mode : fals
e
19/02/21 16:07:06 INFO mapreduce.Job:  map 0% reduce 0%
19/02/21 16:07:11 INFO mapreduce.Job:  map 100% reduce 0%
19/02/21 16:07:15 INFO mapreduce.Job:  map 100% reduce 100%
19/02/21 16:07:16 INFO mapreduce.Job: Job job_1550718539510_0009 completed successfully
19/02/21 16:07:16 INFO mapreduce.Job: Counters: 49
        File System Counters
```

The End of the Processing Log

```
        Map-Reduce Framework
                Map input records=7
                Map output records=7
                Map output bytes=77
                Map output materialized bytes=103
                Input split bytes=218
                Combine input records=0
                Combine output records=0
                Reduce input groups=1
                Reduce shuffle bytes=103
                Reduce input records=7
                Reduce output records=1
                Spilled Records=14
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=406
                CPU time spent (ms)=1210
                Physical memory (bytes) snapshot=518574080
                Virtual memory (bytes) snapshot=6386831360
                Total committed heap usage (bytes)=289406976
        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=1418
        File Output Format Counters
                Bytes Written=9
19/02/21 16:07:16 INFO streaming.StreamJob: Output directory: /stsci5065/maxtemp-output
```

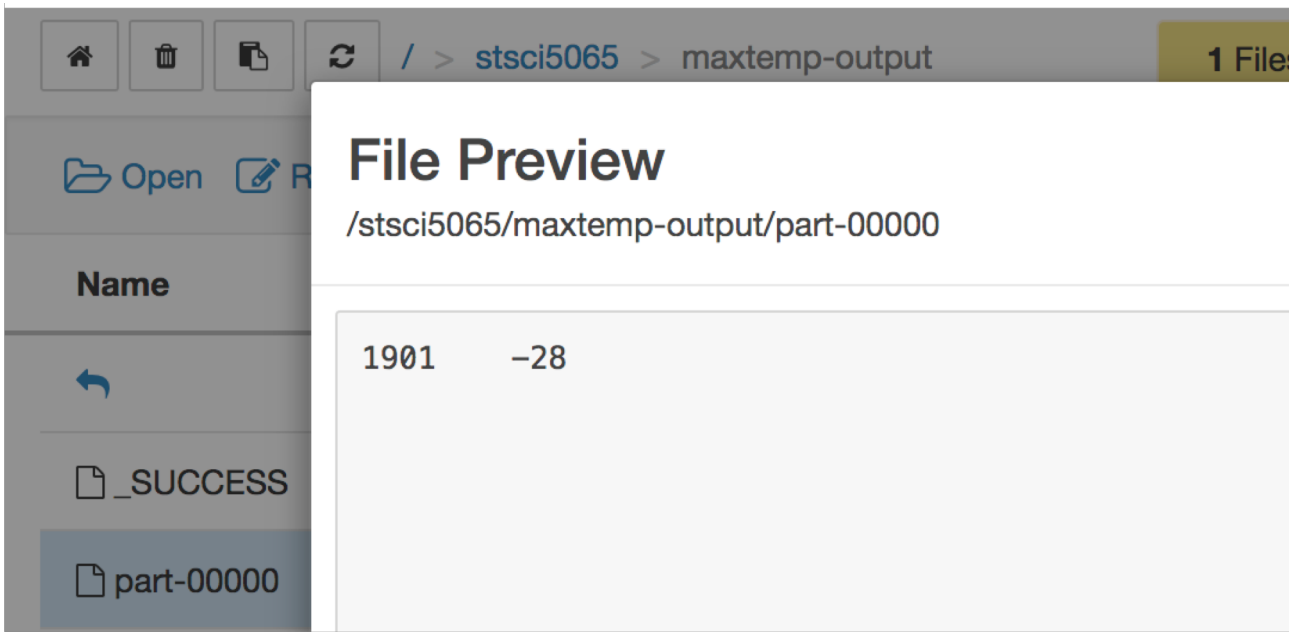
Check the Output

- The output was written to a file called **part-00000** in the **maxtemp-output** directory in the HDFS.

```
[root@sandbox ~]# hadoop fs -cat /stsci5065/maxtemp-output/part-00000
```

```
1901    -28
```

- You may also use Ambari's Files View to display this file



Apply Hadoop Streaming to the 1901-1902 File, a Single File

```
hadoop jar /usr/hdp/2.5.0.0-1245/ \
hadoop-mapreduce/hadoop-streaming.jar  \
-file temp_map.py \
-file temp_reduce.py \
-mapper temp_map.py \
-reducer temp_reduce.py \
-input /stsci5065/data/1901-1902\
-output /stsci5065/maxtemp-output1
```

```
-rw-r--r--   1 admin hdfs     888978 2019-02-21 06:10 /stsci5065/data/1902
-rw-r--r--   1 admin hdfs        282 2019-02-21 05:44 /stsci5065/data/1902-2
-rw-r--r--   1 admin hdfs        403 2019-02-21 15:41 /stsci5065/data/poem.txt
-rw-r--r--   1 admin hdfs        806 2019-02-21 15:41 /stsci5065/data/poem2.txt
-rw-r--r--   1 admin hdfs    5465397 2019-02-21 15:42 /stsci5065/data/shakespeare.txt
-rw-r--r--   1 admin hdfs     566286 2019-02-21 15:43 /stsci5065/data/tom_sawyer.txt
drwxr-xr-x   - admhadoop fs -ls /stsci5065/maxtemp-output1/part-00000
-rw-r--r--   1 root hdfs         18 2019-02-21 16:16 /stsci5065/maxtemp-output1/part-00000
[root@sandbox ~]# hadoop fs -cat /stsci5065/maxtemp-output1/part-00000
1901    317
1902    244
```

Apply Hadoop Streaming to all the files in a directory, 1901 and 1902 Files: the Same Results

```
hadoop jar /usr/hdp/2.5.0.0-1245/ \
hadoop-mapreduce/hadoop-streaming.jar  \
-file temp_map.py \
-file temp_reduce.py \
-mapper temp_map.py \
-reducer temp_reduce.py \
-input /stsci5065/data/two-files\
-output /stsci5065/maxtemp-output2
```

```
[root@sandbox stsci5065L]# hadoop fs -ls /stsci5065/data/two-files/
Found 2 items
-rw-r--r--   1 root hdfs     888190 2019-02-21 15:51 /stsci5065/data/two-files/1901
-rw-r--r--   1 root hdfs     888978 2019-02-21 15:51 /stsci5065/data/two-files/1902
[root@sandbox stsci5065L]# hadoop fs -cat /stsci5065/maxtemp-output2/part-00000
1901    317
1902    244
```

Another Example: Word Counting

- In this example, using Python programs (the mapper and reducer), we read some text files and count how often the words occur.
- The input is a text file(s).
- The output is a text file(s), each line of which contains a word and the count of how many times it occurred, separated by a tab.

The Mapper

WDmapper.py

```
1  #!/usr/bin/env python
2
3  import sys
4
5  # input from STDIN
6  for line in sys.stdin:
7      line = line.strip()
8      words = line.split()
9      for word in words:
10         # the output values are tab-delimited
11         # the word count is 1 since each for
12         # loop only reads one word
13         print '%s\t%s' % (word, 1)
```

The Shuffle

- A lot happens between the map and reduce steps that is largely transparent to the developer.
- In brief, the output of the mappers is transformed and distributed to the reducers (termed the shuffle step) in such a way that
  - All key/value pairs are sorted before being presented to the reduce function.
  - All key/value pairs sharing the same key are sent to the same reducer.

The Reducer

WDreducer.py

```
 1 #!/usr/bin/env python
 2
 3 import sys
 4
 5 current_word = None
 6 current_count = 0
 7 word = None
 8
 9 # input comes from STDIN
10 for line in sys.stdin:
11     line = line.strip()
12
13     # parse the input we got from WDmapper.py
14     word, count = line.split('\t', 1)
15
16     # convert count (a string) to int
17     try:
18         count = int(count)
19     except ValueError:
20         # if count was not a number, silently ignore the line
21         continue
```

The Reducer (cont'd)

WDreducer.py

```
22
23      # Hadoop sorts map output by key (here: word) before it
24      # is passed to the reducer
25      if current_word == word:
26          current_count += count
27      else:
28          if current_word:
29              # write result to STDOUT
30              print '%s\t%s' % (current_word, current_count)
31          current_count = count
32          current_word = word
33
34  # output the last word if needed!
35  if current_word == word:
36      print '%s\t%s' % (current_word, current_count)
```

Add the Execution Permission to Files

- The original file permissions of WDmapper.py and WDreducer.py are rw-r--r--.
- The execution permission needs to be added to the files. This time we use the following commands:

  chmod +x /stsci5065/WDmapper.py

  chmod +x /stsci5065/WDreducer.py

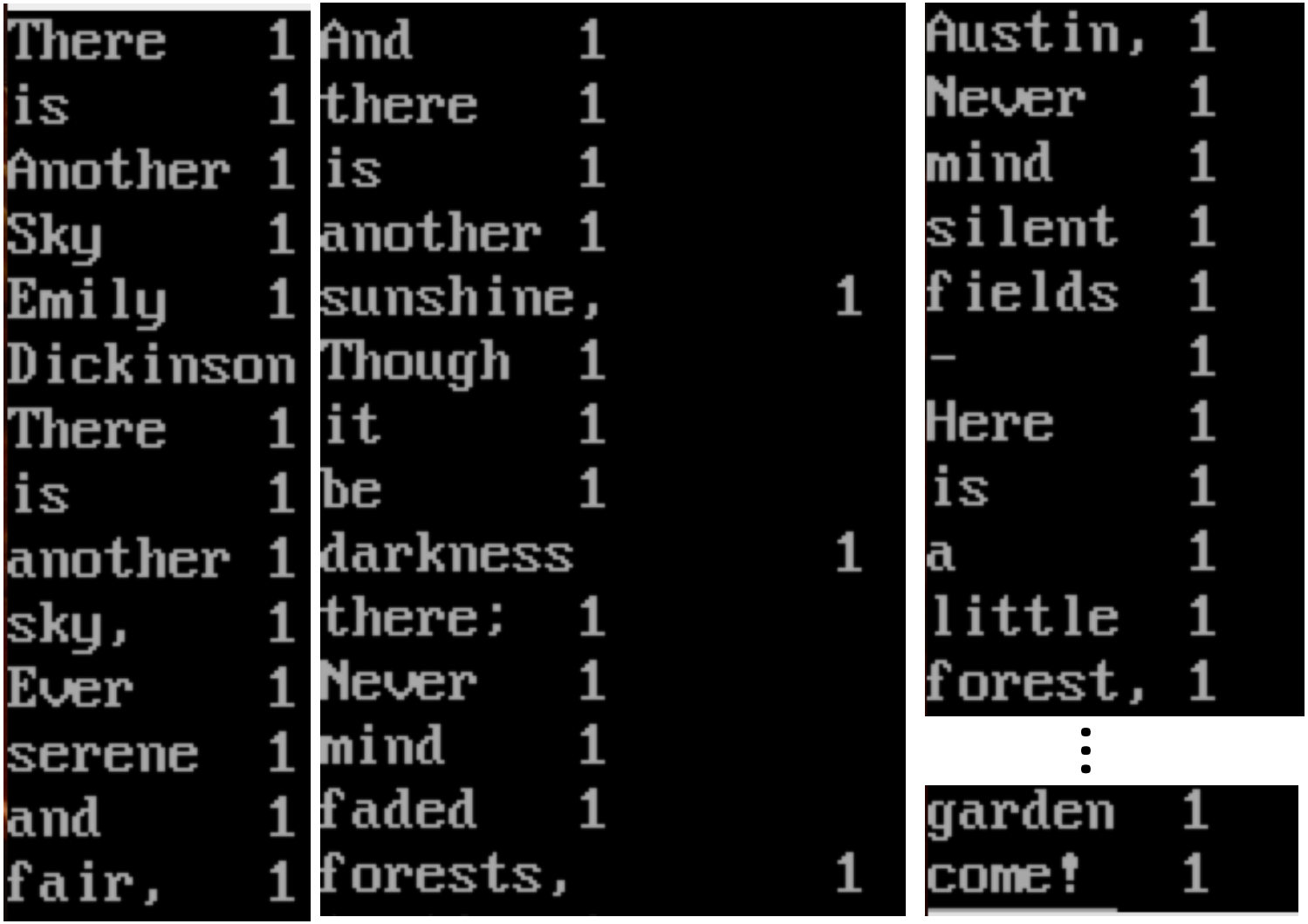
- Their file permissions became rwxr-xr-x.

```
-rwxr-xr-x 1 root root     315 Feb 21 15:22 WDmapper.py
-rwxr-xr-x 1 root root     875 Feb 21 15:36 WDreducer.py
```

First Test Your Code Locally

- Test the mapper, WDmapper.py

  cat ./poem.txt | ./WDmapper.py



First Test Your Code Locally

Test the reducer, WDreducer.py with the mapper

cat ./poem.txt | ./WDmapper.py | sort | ./WDreducer.py

```
-       1
a       3
and     1
And     1
another 2
Another 1
Austin, 1
be      1
bee     1
been;   1
bright  1
brighter        1
brother,        1
come!   1
darkness        1
```

...

```
not     1
Prithee,        1
serene  1
silent  1
sky,    1
Sky     1
sunshine,       1
the     1
there;  1
there   1
There   2
Though  1
unfading        1
Where   1
Whose   1
```

Hadoop Streaming on Shakespeare.txt

```
hadoop jar /usr/hdp/2.5.0.0-1245/ \
hadoop-mapreduce/hadoop-streaming.jar \
-file WDmapper.py \
-file WDreducer.py \
-mapper WDmapper.py \
-reducer WDreducer.py \
-input /stsci5065/data/shakespeare.txt\
-output /stsci5065/maxtemp-output3
```

The Streaming Results

```
[root@sandbox stsci5065L]# hadoop jar /usr/hdp/2.5.0.0-1245/hadoop-mapreduce/hadoop-streaming.jar  -file W
Dmapper.py  -file WDreducer.py  -mapper WDmapper.py  -reducer WDreducer.py  -input /stsci5065/data/shakesp
eare.txt -output /stsci5065/maxtemp-output3
19/02/21 16:50:55 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files i
nstead.
packageJobJar: [WDmapper.py, WDreducer.py] [/usr/hdp/2.6.1.0-129/hadoop-mapreduce/hadoop-streaming-2.7.3.2
.6.1.0-129.jar] /tmp/streamjob8252519509295690169.jar tmpDir=null
19/02/21 16:50:56 INFO client.RMProxy: Connecting to ResourceManager at sandbox.hortonworks.com/172.17.0.2
:8032
19/02/21 16:50:56 INFO client.AHSProxy: Connecting to Application History server at sandbox.hortonworks.co
m/172.17.0.2:10200
19/02/21 16:50:57 INFO client.RMProxy: Connecting to ResourceManager at sandbox.hortonworks.com/172.17.0.2
:8032
19/02/21 16:50:57 INFO client.AHSProxy: Connecting to Application History server at sandbox.hortonworks.co
m/172.17.0.2:10200
19/02/21 16:50:57 INFO mapred.FileInputFormat: Total input paths to process : 1
19/02/21 16:50:58 INFO mapreduce.JobSubmitter: number of splits:2
19/02/21 16:50:58 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1550718539510_0015
19/02/21 16:50:58 INFO impl.YarnClientImpl: Submitted application application_1550718539510_0015
19/02/21 16:50:58 INFO mapreduce.Job: The url to track the job: http://sandbox.hortonworks.com:8088/proxy/
application_1550718539510_0015/
19/02/21 16:50:58 INFO mapreduce.Job: Running job: job_1550718539510_0015
19/02/21 16:51:04 INFO mapreduce.Job: Job job_1550718539510_0015 running in uber mode : false
19/02/21 16:51:04 INFO mapreduce.Job:  map 0% reduce 0%
19/02/21 16:51:10 INFO mapreduce.Job:  map 100% reduce 0%
19/02/21 16:51:17 INFO mapreduce.Job:  map 100% reduce 100%
19/02/21 16:51:18 INFO mapreduce.Job: Job job_1550718539510_0015 completed successfully
19/02/21 16:51:18 INFO mapreduce.Job: Counters: 49
```

⋮

```
INFO streaming.StreamJob: Output directory: /stsci5065/maxtemp-output3
```

The Output

```
hadoop fs -cat /stsci5065/maxtemp-output3/part-00000
```

```
"        241
"'Tis    1
"A       4
"AS-IS".         1
"Air,"   1
"Alas,   1
"Amen"   2
"Amen"?  1
"Amen,"  1
"And     1
"Aroint  1
"B       1
"Black   1
"Break   1
"Brutus"         1
"Brutus,         2
```

...

```
zeal,    7
zeal.    4
zealous  6
zeals,   1
zed!     1
zenith   1
zephyrs  1
zir,     1
zir.     1
zo       1
zodiac   1
zodiacs  1
zone,    1
zounds!  1
zounds,  1
zwagger'd        1
```