

## Hadoop Streaming - Average

```
hadoop fs -mkdir /user/ec2-user/lineorder
```

```
hadoop fs -put /home/ec2-user /user/ec2-user/lineorder
```

```
find . -name "hadoop-streaming-2.6.4.jar" -print  
./hadoop-2.6.4/share/hadoop/tools/lib/hadoop-streaming-2.6.4.jar
```

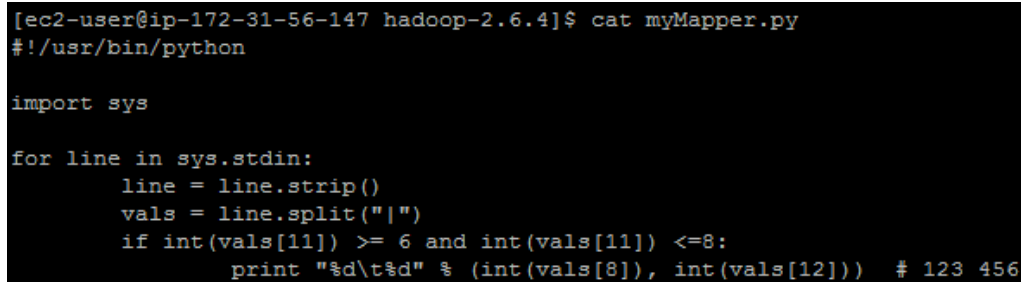
```
cp ./share/hadoop/tools/lib/hadoop-streaming-2.6.4.jar .
```

### myMapper.py

```
#!/usr/bin/python
```

```
import sys
```

```
for line in sys.stdin:  
    line = line.strip()  
    vals = line.split("|")  
    if int(vals[11]) >= 6 and int(vals[11]) <=8:  
        print "%d\t%d" % (int(vals[8]), int(vals[12])) # 123 456
```



```
[ec2-user@ip-172-31-56-147 hadoop-2.6.4]$ cat myMapper.py  
#!/usr/bin/python  
  
import sys  
  
for line in sys.stdin:  
    line = line.strip()  
    vals = line.split("|")  
    if int(vals[11]) >= 6 and int(vals[11]) <=8:  
        print "%d\t%d" % (int(vals[8]), int(vals[12])) # 123 456
```

### myReducer.py

```
#!/usr/bin/python
```

```
import sys
```

```
curr_id = None  
curr_cnt = 0  
id = None  
val = 0  
# The input comes from standard input (line by line)  
for line in sys.stdin:  
    line = line.strip()  
    # parse the line and split it by '\t'  
    ln = line.split('\t') # [1, 5]  
    # grab the key  
    id = int(ln[0]) # current received key is lo_quantity  
    if curr_id == id:  
        curr_cnt += 1  
        val = val + int(ln[1])  
    else:
```

```

if curr_id: # output the count, single key completed
    # NOTE: Change this to '%s\t%d' if your key is a string
    print '%d\t%d' % (curr_id, (val/curr_cnt)) # print 1\t2 if you saw two 1s
curr_id = id # Reset the current key to the new key (e.g., 6)
curr_cnt = 1
val = 0
val = val + int(ln[1])

```

# output the last key

```

if curr_id == id:
    print '%d\t%d' % (curr_id, (val/curr_cnt))

```

```

[ec2-user@ip-172-31-56-147 hadoop-2.6.4]$ cat myReducer.py
#!/usr/bin/python
import sys

curr_id = None
curr_cnt = 0
id = None
val = 0
# The input comes from standard input (line by line)
for line in sys.stdin:
    line = line.strip()
    # parse the line and split it by '\t'
    ln = line.split('\t') # [1, 5]
    # grab the key
    id = int(ln[0]) # current received key is lo_quantity
    if curr_id == id:
        curr_cnt += 1
        val = val + int(ln[1])
    else:
        if curr_id: # output the count, single key completed
            # NOTE: Change this to '%s\t%d' if your key is a string
            print '%d\t%d' % (curr_id, (val/curr_cnt)) # print 1\t2 if you saw two 1s
        curr_id = id # Reset the current key to the new key (e.g., 6)
        curr_cnt = 1
        val = 0
        val = val + int(ln[1])

# output the last key
if curr_id == id:
    print '%d\t%d' % (curr_id, (val/curr_cnt))

```

```
hadoop jar hadoop-streaming-2.6.4.jar -input /user/ec2-user/lineorder/lineorder.tbl -output  
/data/output2 -mapper myMapper.py -reducer myReducer.py -file myReducer.py -file  
myMapper.py
```

```
20/10/15 05:30:54 INFO mapreduce.Job: map 100% reduce 100%  
20/10/15 05:30:54 INFO mapreduce.Job: Job job_1602738558497_0004 completed successfully  
20/10/15 05:30:55 INFO mapreduce.Job: Counters: 50  
  File System Counters  
    FILE: Number of bytes read=20742636  
    FILE: Number of bytes written=42145343  
    FILE: Number of read operations=0  
    FILE: Number of large read operations=0  
    FILE: Number of write operations=0  
    HDFS: Number of bytes read=594329915  
    HDFS: Number of bytes written=534  
    HDFS: Number of read operations=18  
    HDFS: Number of large read operations=0  
    HDFS: Number of write operations=2  
  Job Counters  
    Killed map tasks=1  
    Launched map tasks=6  
    Launched reduce tasks=1  
    Data-local map tasks=6  
    Total time spent by all maps in occupied slots (ms)=305118  
    Total time spent by all reduces in occupied slots (ms)=25423  
    Total time spent by all map tasks (ms)=305118  
    Total time spent by all reduce tasks (ms)=25423  
    Total vcore-milliseconds taken by all map tasks=305118  
    Total vcore-milliseconds taken by all reduce tasks=25423  
    Total megabyte-milliseconds taken by all map tasks=312440832  
    Total megabyte-milliseconds taken by all reduce tasks=26033152  
  Map-Reduce Framework  
    Map input records=6001215  
    Map output records=1635965  
    Map output bytes=17470700  
    Map output materialized bytes=20742660  
    Input split bytes=530  
    Combine input records=0  
    Combine output records=0  
    Reduce input groups=50  
    Reduce shuffle bytes=20742660  
    Reduce input records=1635965  
    Reduce output records=50  
    Spilled Records=3271930  
    Shuffled Maps =5  
    Failed Shuffles=0  
    Merged Map outputs=5  
    GC time elapsed (ms)=2203  
    CPU time spent (ms)=34850  
    Physical memory (bytes) snapshot=1075802112  
    Virtual memory (bytes) snapshot=12619886592  
    Total committed heap usage (bytes)=741675008  
  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=594329385  
  File Output Format Counters  
    Bytes Written=534  
20/10/15 05:30:55 INFO streaming.StreamJob: Output directory: /data/output5
```

**Output:**

```
[ec2-user@ip-172-31-56-147 hadoop-2.6.4]$ hadoop fs -cat /data/output5/part-00000
```

```
1    139452
10   1394592
11   1533974
12   1672518
13   1813827
14   1950263
15   2090607
16   2230922
17   2371535
18   2516921
19   2653429
2    278553
20   2789022
21   2925041
22   3071083
23   3202331
24   3345892
25   3485642
26   3634084
27   3768283
28   3905890
29   4048916
3    418738
30   4188699
31   4324416
32   4471392
33   4602964
34   4739361
35   4880643
36   5026968
37   5164615
38   5299180
39   5439082
4    557755
40   5577892
41   5720788
42   5858283
43   6000194
44   6135599
45   6271572
46   6434834
47   6555244
48   6699379
49   6837399
5    696104
50   6966732
6    837641
7    977779
```

8 1115301  
9 1254213

```
[ec2-user@ip-172-31-56-147 hadoop-2.6.4]$ hadoop fs -cat /data/output5/part-00000
1      139452
10     1394592
11     1533974
12     1672518
13     1813827
14     1950263
15     2090607
16     2230922
17     2371535
18     2516921
19     2653429
2      278553
20     2789022
21     2925041
22     3071083
23     3202331
24     3345892
25     3485642
26     3634084
27     3768283
28     3905890
29     4048916
3      418738
30     4188699
31     4324416
32     4471392
33     4602964
34     4739361
35     4880643
36     5026968
37     5164615
38     5299180
39     5439082
4      557755
40     5577892
41     5720788
42     5858283
43     6000194
44     6135599
45     6271572
46     6434834
47     6555244
48     6699379
49     6837399
5      696104
50     6966732
6      837641
7      977779
8      1115301
9      1254213
```