Hadoop Streaming – Standard Deviation

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

hadoop fs -put /home/ec2-user/lineorder.tbl /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 .

<u>myMapper.py</u>

```
#!/usr/bin/python

import sys

for line in sys.stdin:
    line = line.strip()
    vals = line.split("|")
    if int(vals[8]) >= 15 and int(vals[8]) <=18:
        print "%s\t%d" % (vals[16], int(vals[14]))  # 123 456
```

```
[ec2-user@ip-172-31-75-50 ~]$ cat myMapper.py
#!/usr/bin/python

import sys

for line in sys.stdin:
        line = line.strip()
        vals = line.split("|")
        if int(vals[8]) >= 15 and int(vals[8]) <=18:
                print "%s\t%d" % (vals[16], int(vals[14]))  # 123 456
```

<u>myReducer.py</u>

```
#!/usr/bin/python
import sys

curr_id = None
curr_cnt = 0
id = None
val = []
avg = 0
variance = []
# 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 = ln[0]   # current received key is lo_quantity
```

```
    if curr_id == id:
        curr_cnt += 1
        val.append(int(ln[1]))
    else:
        if curr_id: # output the count, single key completed
            avg = sum(val) * 1.0 / len(val)
            variance = list(map( lambda x: (x - avg)**2, val))
            print '%s\t%f' % (curr_id, (sum(variance) * 1.0 / len(variance)) ** 0.5)
        curr_id = id
        curr_cnt = 1
        val = []
        avg = 0
        variance = []
        val.append(int(ln[1]))

# output the last key
if curr_id == id:
    avg = sum(val) * 1.0 / len(val)
    variance = list(map( lambda x: (x - avg)**2, val))
    print '%s\t%f' % (curr_id, (sum(variance) * 1.0 / len(variance)) ** 0.5)
```

```
[ec2-user@ip-172-31-75-50 ~]$ cat myReducer.py
#!/usr/bin/python
import sys

curr_id = None
curr_cnt = 0
id = None
val = []
avg = 0
variance = []
# 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 = ln[0]   # current received key is lo_quantity
    if curr_id == id:
        curr_cnt += 1
        val.append(int(ln[1]))
    else:
        if curr_id: # output the count, single key completed
            avg = sum(val) * 1.0 / len(val)
            variance = list(map( lambda x: (x - avg)**2, val))
            print '%s\t%f' % (curr_id, (sum(variance) * 1.0 / len(variance)) ** 0.5)
        curr_id = id
        curr_cnt = 1
        val = []
        avg = 0
        variance = []
        val.append(int(ln[1]))

# output the last key
if curr_id == id:
    avg = sum(val) * 1.0 / len(val)
    variance = list(map( lambda x: (x - avg)**2, val))
    print '%s\t%f' % (curr_id, (sum(variance) * 1.0 / len(variance)) ** 0.5)
```

hadoop-2.6.4]$ hadoop jar hadoop-streaming-2.6.4.jar -input /user/ec2-user/lineorder/lineorder.tbl -output /data/output3 -mapper ../myMapper.py -reducer ../myReducer.py -file ../myReducer.py -file ../myMapper.py

```
                Launched reduce tasks=1
                Data-local map tasks=6
                Total time spent by all maps in occupied slots (ms)=53103
                Total time spent by all reduces in occupied slots (ms)=6328
                Total time spent by all map tasks (ms)=53103
                Total time spent by all reduce tasks (ms)=6328
                Total vcore-milliseconds taken by all map tasks=53103
                Total vcore-milliseconds taken by all reduce tasks=6328
                Total megabyte-milliseconds taken by all map tasks=54377472
                Total megabyte-milliseconds taken by all reduce tasks=6479872
        Map-Reduce Framework
                Map input records=6001215
                Map output records=480357
                Map output bytes=3500084
                Map output materialized bytes=4460828
                Input split bytes=545
                Combine input records=0
                Combine output records=0
                Reduce input groups=7
                Reduce shuffle bytes=4460828
                Reduce input records=480357
                Reduce output records=7
                Spilled Records=960714
                Shuffled Maps =5
                Failed Shuffles=0
                Merged Map outputs=5
                GC time elapsed (ms)=657
                CPU time spent (ms)=17540
                Physical memory (bytes) snapshot=1475264512
                Virtual memory (bytes) snapshot=12757639168
                Total committed heap usage (bytes)=1077936128
        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=100
20/10/23 21:41:47 INFO streaming.StreamJob: Output directory: /data/output3
```

Output:

hadoop fs -cat /data/output3/part-00000

```
[ec2-user@ip-172-31-75-50 hadoop-2.6.4]$ hadoop fs -cat /data/output3/part-00000
AIR     2.585300
FOB     2.580486
MAIL    2.576278
RAIL    2.572036
REG AIR 2.578802
SHIP    2.579333
TRUCK   2.583316
```