Assignment -3.1

1. **Task1:** Execute **WordMedian, WordMean, WordStandardDeviation** programs using hadoop-mapreduce-examples-2.9.0.jar file present in acadgild VM.

Answer 1:

I will be using **hadoop-mapreduce-examples-2.6.5.jar** file as this is the available version of examples in my acadgild VM.

JAR file: **hadoop-mapreduce-examples-2.6.5.jar**

JAR file Location: **/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce**

Now first Get into the location of JAR file using the below command as shown in the screenshot (refer Fig 3.1).

cd /home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce

```
[acadgild@localhost ~]$ cd /home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce
[acadgild@localhost mapreduce]$
```

Fig 3.1

Here we can see the cursor has been placed inside the “mapreduce” directory.

Please find the input file details which will be used for all three scenarios - WordMedian, WordMean, WordStandardDeviation.

InputFile HDFS Location: **/user/acadgild/hadoop/word-count.txt**

Input file content can be viewed using **cat command (Refer Fig 3.2).**

Note: This file is placed in HDFS location as mapreduce functions can be implemented only on HDFS.

```
[acadgild@localhost ~]$ hadoop fs -cat /user/acadgild/hadoop/word-count.txt
18/02/25 11:11:43 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where a
pplicable
It’s 2018, but still tough to get online in the Andamans

Visitors from the mainland are at first perplexed and then frustrated when they cannot ‘stay connected’ in the Andaman and Nicobar Islands
. A strong Internet connection is rare here, data services for smartphones are almost non-existent even in Port Blair and voice calls drop
 frequently. Islanders face difficulty in banking and buying online, and GST returns are often filed late.

Poor connections can potentially be disastrous. In October 2017, a bus with 39 students on its way to Billyground from a college in Mayabu
nder was gutted in a fire. There were no casualties, but Fire Services personnel reached late because mobile phones did not work at the si
te.

“I have been staying at Diglipur since January 2017. Internet is almost non-existent and even the phone network doesn’t work for more than
 15 days in a month,” says Dr. Punam Tripathi, author of Routledge’s forthcoming book, The Vulnerable Andaman and Nicobar Islands: A Study
 of Disasters and Response. The National Optical Fibre Network (NOFN), envisioned to cover 26 States and Union Territories in 2011, is yet
 to connect the Andaman islands, which rely on expensive satellite bandwidth. “Do you have BSNL?” is thus a frequently heard query. BSNL s
ources its bandwidth from the Indian Space Research Organisation’s GSAT 16 and GSAT 18 satellites. It has hired 24 transponders for 72 bas
e transceiver stations (BTS) for 3G and 160 for 2G across the islands, and also has 52 landline exchanges and 480 leased circuits.

Landline-linked broadband Internet is the most reliable data service here. Government authorities, banks and institutional users get 2 Mbp
s leased VSAT (very small aperture terminal) Internet lines. WhatsApp does work in areas where 3G coverage is not available, but is a lot
slower. A plan to nearly double satellite bandwidth to 2 Gbps was approved by the Department of Telecommunications, but expansion has been
 hit by problems like unsuitability of old technology. Approximately 10% of the 1,300 MHz bandwidth that BSNL gets from ISRO is ‘lost in t
```

Fig 3.2

Now to get the description of each of the functions available in the JAR file execute the JAR file with the below command (Refer Fig 3.3).

hadoop jar hadoop-mapreduce-examples-2.6.5.jar

```
[acadgild@localhost mapreduce]$ hadoop jar hadoop-mapreduce-examples-2.6.5.jar
An example program must be given as the first argument.
Valid program names are:
  aggregatewordcount: An Aggregate based map/reduce program that counts the words in the input files.
  aggregatewordhist: An Aggregate based map/reduce program that computes the histogram of the words in the input files.
  bbp: A map/reduce program that uses Bailey-Borwein-Plouffe to compute exact digits of Pi.
  dbcount: An example job that count the pageview counts from a database.
  distbbp: A map/reduce program that uses a BBP-type formula to compute exact bits of Pi.
  grep: A map/reduce program that counts the matches of a regex in the input.
  join: A job that effects a join over sorted, equally partitioned datasets
  multifilewc: A job that counts words from several files.
  pentomino: A map/reduce tile laying program to find solutions to pentomino problems.
  pi: A map/reduce program that estimates Pi using a quasi-Monte Carlo method.
  randomtextwriter: A map/reduce program that writes 10GB of random textual data per node.
  randomwriter: A map/reduce program that writes 10GB of random data per node.
  secondarysort: An example defining a secondary sort to the reduce.
  sort: A map/reduce program that sorts the data written by the random writer.
  sudoku: A sudoku solver.
  teragen: Generate data for the terasort
  terasort: Run the terasort
  teravalidate: Checking results of terasort
  wordcount: A map/reduce program that counts the words in the input files.
  wordmean: A map/reduce program that counts the average length of the words in the input files.
  wordmedian: A map/reduce program that counts the median length of the words in the input files.
  wordstandarddeviation: A map/reduce program that counts the standard deviation of the length of the words in the input files.
[acadgild@localhost mapreduce]$
```

Fig 3.3

Here the descriptions of all three functions are given below.

wordmean: A map/reduce program that counts the average length of the words in the input files.

wordmedian: A map/reduce program that counts the median length of the words in the input files.

wordstandarddeviation: A map/reduce program that counts the standard deviation of the length of the words in the input files.

1. WordMean: Use below command to execute this function with above raw materials.
   Input: (refer Fig 3.4)
   hadoop jar hadoop-mapreduce-examples-2.6.5.jar wordmean /user/acadgild/hadoop/word-count.txt /user/acadgild/hadoop/wordmeanOut

```
[acadgild@localhost mapreduce]$ hadoop jar hadoop-mapreduce-examples-2.6.5.jar wordmean /user/acadgild/hadoop/word-count.txt /user/acadgil
d/hadoop/wordmeanOut
18/02/25 12:20:27 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where a
pplicable
18/02/25 12:20:29 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/02/25 12:20:31 INFO input.FileInputFormat: Total input paths to process : 1
18/02/25 12:20:31 INFO mapreduce.JobSubmitter: number of splits:1
18/02/25 12:20:31 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1519536798138_0001
18/02/25 12:20:32 INFO impl.YarnClientImpl: Submitted application application_1519536798138_0001
18/02/25 12:20:32 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1519536798138_0001/
18/02/25 12:20:32 INFO mapreduce.Job: Running job: job_1519536798138_0001
18/02/25 12:20:43 INFO mapreduce.Job: Job job_1519536798138_0001 running in uber mode : false
18/02/25 12:20:43 INFO mapreduce.Job:  map 0% reduce 0%
18/02/25 12:20:51 INFO mapreduce.Job:  map 100% reduce 0%
18/02/25 12:20:59 INFO mapreduce.Job:  map 100% reduce 100%
18/02/25 12:20:59 INFO mapreduce.Job: Job job_1519536798138_0001 completed successfully
18/02/25 12:20:59 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=39
                FILE: Number of bytes written=215477
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=2257
                HDFS: Number of bytes written=22
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=4771
                Total time spent by all reduces in occupied slots (ms)=5205
                Total time spent by all map tasks (ms)=4771
                Total time spent by all reduce tasks (ms)=5205
                Total vcore-milliseconds taken by all map tasks=4771
                Total vcore-milliseconds taken by all reduce tasks=5205
```

Fig 3.4

Output: (refer Fig 3.5)

The mean is: 5.178885630498534

```
        Map-Reduce Framework
                Map input records=9
                Map output records=682
                Map output bytes=9889
                Map output materialized bytes=39
                Input split bytes=122
                Combine input records=682
                Combine output records=2
                Reduce input groups=2
                Reduce shuffle bytes=39
                Reduce input records=2
                Reduce output records=2
                Spilled Records=4
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=121
                CPU time spent (ms)=1420
                Physical memory (bytes) snapshot=321667072
                Virtual memory (bytes) snapshot=4118245376
                Total committed heap usage (bytes)=222429184
        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=2135
        File Output Format Counters
                Bytes Written=22
The mean is: 5.178885630498534
[acadgild@localhost mapreduce]$
```

Fig 3.5

Also refer the output file in the HDFS location to get the actual counts (refer Fig 3.6) - **/user/acadgild/hadoop/wordmeanOut/part-r-00000**

```
[acadgild@localhost ~]$ hadoop fs -cat /user/acadgild/hadoop/word-count.txt
18/02/25 11:11:43 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where a
pplicable
It's 2018, but still tough to get online in the Andamans

Visitors from the mainland are at first perplexed and then frustrated when they cannot 'stay connected' in the Andaman and Nicobar Islands
. A strong Internet connection is rare here, data services for smartphones are almost non-existent even in Port Blair and voice calls drop
 frequently. Islanders face difficulty in banking and buying online, and GST returns are often filed late.

Poor connections can potentially be disastrous. In October 2017, a bus with 39 students on its way to Billyground from a college in Mayabu
nder was gutted in a fire. There were no casualties, but Fire Services personnel reached late because mobile phones did not work at the si
te.

"I have been staying at Diglipur since January 2017. Internet is almost non-existent and even the phone network doesn't work for more than
 15 days in a month," says Dr. Punam Tripathi, author of Routledge's forthcoming book, The Vulnerable Andaman and Nicobar Islands: A Study
 of Disasters and Response. The National Optical Fibre Network (NOFN), envisioned to cover 26 States and Union Territories in 2011, is yet
 to connect the Andaman islands, which rely on expensive satellite bandwidth. "Do you have BSNL?" is thus a frequently heard query. BSNL s
ources its bandwidth from the Indian Space Research Organisation's GSAT 16 and GSAT 18 satellites. It has hired 24 transponders for 72 bas
e transceiver stations (BTS) for 3G and 160 for 2G across the islands, and also has 52 landline exchanges and 480 leased circuits.

Landline-linked broadband Internet is the most reliable data service here. Government authorities, banks and institutional users get 2 Mbp
s leased VSAT (very small aperture terminal) Internet lines. WhatsApp does work in areas where 3G coverage is not available, but is a lot
slower. A plan to nearly double satellite bandwidth to 2 Gbps was approved by the Department of Telecommunications, but expansion has been
 hit by problems like unsuitability of old technology. Approximately 10% of the 1,300 MHz bandwidth that BSNL gets from ISRO is 'lost in t
```

Fig 3.6

2. WordMedian: Use below command to execute this function with above raw materials.

   Input: (refer Fig 3.7)

   hadoop jar hadoop-mapreduce-examples-2.6.5.jar wordmedian /user/acadgild/hadoop/word-count.txt /user/acadgild/hadoop/wordmedianOut

```
[acadgild@localhost mapreduce]$ hadoop jar hadoop-mapreduce-examples-2.6.5.jar wordmedian /user/acadgild/hadoop/word-count.txt /user/acadg
ild/hadoop/wordmedianOut
18/02/25 12:57:06 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where a
pplicable
18/02/25 12:57:07 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/02/25 12:57:08 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and e
xecute your application with ToolRunner to remedy this.
18/02/25 12:57:08 INFO input.FileInputFormat: Total input paths to process : 1
18/02/25 12:57:08 INFO mapreduce.JobSubmitter: number of splits:1
18/02/25 12:57:09 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1519536798138_0002
18/02/25 12:57:09 INFO impl.YarnClientImpl: Submitted application application_1519536798138_0002
18/02/25 12:57:09 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1519536798138_0002/
18/02/25 12:57:09 INFO mapreduce.Job: Running job: job_1519536798138_0002
18/02/25 12:57:18 INFO mapreduce.Job: Job job_1519536798138_0002 running in uber mode : false
18/02/25 12:57:18 INFO mapreduce.Job:  map 0% reduce 0%
18/02/25 12:57:25 INFO mapreduce.Job:  map 100% reduce 0%
18/02/25 12:57:32 INFO mapreduce.Job:  map 100% reduce 100%
18/02/25 12:57:32 INFO mapreduce.Job: Job job_1519536798138_0002 completed successfully
18/02/25 12:57:33 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=166
                FILE: Number of bytes written=215471
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=2257
                HDFS: Number of bytes written=82
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=4831
                Total time spent by all reduces in occupied slots (ms)=4559
                Total time spent by all map tasks (ms)=4831
```

Fig 3.7

Output: (refer Fig 3.8)

The median is: 4

```
        Map-Reduce Framework
                Map input records=9
                Map output records=341
                Map output bytes=2728
                Map output materialized bytes=166
                Input split bytes=122
                Combine input records=341
                Combine output records=16
                Reduce input groups=16
                Reduce shuffle bytes=166
                Reduce input records=16
                Reduce output records=16
                Spilled Records=32
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=113
                CPU time spent (ms)=1470
                Physical memory (bytes) snapshot=321150976
                Virtual memory (bytes) snapshot=4118245376
                Total committed heap usage (bytes)=222429184
        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=2135
        File Output Format Counters
                Bytes Written=82
The median is: 4
[acadgild@localhost mapreduce]$
```

Fig 3.8

Also refer the output file in the HDFS location to get the actual counts (refer Fig 3.9) - **/user/acadgild/hadoop/wordmedianOut/part-r-00000**

```
[acadgild@localhost mapreduce]$ hadoop fs -cat /user/acadgild/hadoop/wordmedianOut/part-r-00000
18/02/25 13:03:01 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where a
pplicable
1       11
2       52
3       66
4       50
5       40
6       20
7       24
8       25
9       17
10      12
11      13
12      4
13      3
14      2
15      1
19      1
```

Fig 3.9

3. WordStandarDeviation: Use below command to execute this function with above raw materials.

   Input: (refer Fig 3.10)

   hadoop jar hadoop-mapreduce-examples-2.6.5.jar wordstandarddeviation /user/acadgild/hadoop/word-count.txt /user/acadgild/hadoop/wordstandarddeviationOut

```
[acadgild@localhost mapreduce]$ hadoop jar hadoop-mapreduce-examples-2.6.5.jar wordstandarddeviation /user/acadgild/hadoop/word-count.txt
/user/acadgild/hadoop/wordstandarddeviationOut
18/02/25 13:16:01 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where a
pplicable
18/02/25 13:16:02 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/02/25 13:16:03 INFO input.FileInputFormat: Total input paths to process : 1
18/02/25 13:16:03 INFO mapreduce.JobSubmitter: number of splits:1
18/02/25 13:16:03 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1519536798138_0003
18/02/25 13:16:04 INFO impl.YarnClientImpl: Submitted application application_1519536798138_0003
18/02/25 13:16:04 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1519536798138_0003/
18/02/25 13:16:04 INFO mapreduce.Job: Running job: job_1519536798138_0003
18/02/25 13:16:12 INFO mapreduce.Job: Job job_1519536798138_0003 running in uber mode : false
18/02/25 13:16:12 INFO mapreduce.Job:  map 0% reduce 0%
18/02/25 13:16:19 INFO mapreduce.Job:  map 100% reduce 0%
18/02/25 13:16:26 INFO mapreduce.Job:  map 100% reduce 100%
18/02/25 13:16:26 INFO mapreduce.Job: Job job_1519536798138_0003 completed successfully
18/02/25 13:16:26 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=56
                FILE: Number of bytes written=215697
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=2257
                HDFS: Number of bytes written=35
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=4169
                Total time spent by all reduces in occupied slots (ms)=4598
                Total time spent by all map tasks (ms)=4169
                Total time spent by all reduce tasks (ms)=4598
                Total vcore-milliseconds taken by all map tasks=4169
                Total vcore-milliseconds taken by all reduce tasks=4598
```

Fig 3.10

Output: (refer Fig 3.11)

The standard deviation is: 3.050930330945026

```
        Map-Reduce Framework
                Map input records=9
                Map output records=1023
                Map output bytes=15004
                Map output materialized bytes=56
                Input split bytes=122
                Combine input records=1023
                Combine output records=3
                Reduce input groups=3
                Reduce shuffle bytes=56
                Reduce input records=3
                Reduce output records=3
                Spilled Records=6
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=104
                CPU time spent (ms)=1310
                Physical memory (bytes) snapshot=321138688
                Virtual memory (bytes) snapshot=4118241280
                Total committed heap usage (bytes)=222429184
        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=2135
        File Output Format Counters
                Bytes Written=35
The standard deviation is: 3.050930330945026
```

Fig 3.11

Also refer the output file in the HDFS location to get the actual counts (refer Fig 3.12) -

/user/acadgild/hadoop/wordstandarddeviationOut/part-r-00000

```
[acadgild@localhost mapreduce]$ hadoop fs -cat /user/acadgild/hadoop/wordstandarddeviationOut/part-r-00000
18/02/25 13:27:47 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where a
pplicable
count   341
length  1766
square  12320
```

Fig 3.12