CSE488 (Section 1)

[Spring 2023]

Lab Assignment Submission Report

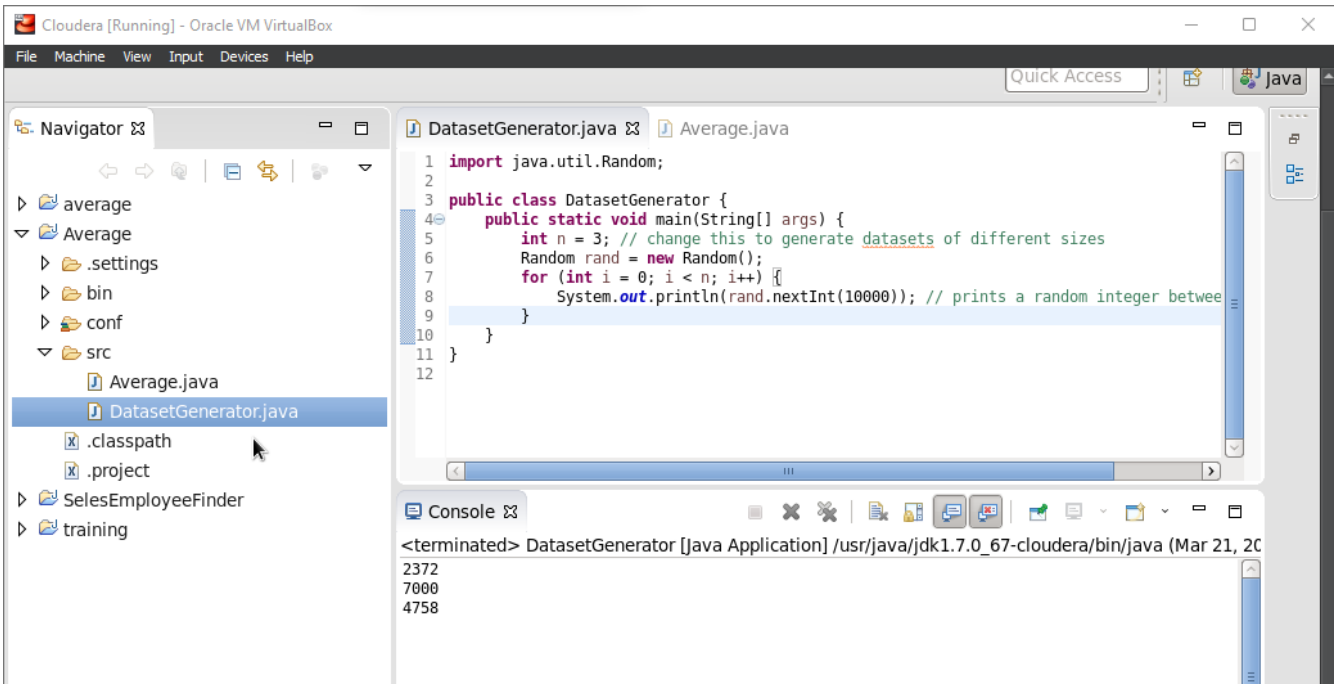
Offline Assignment

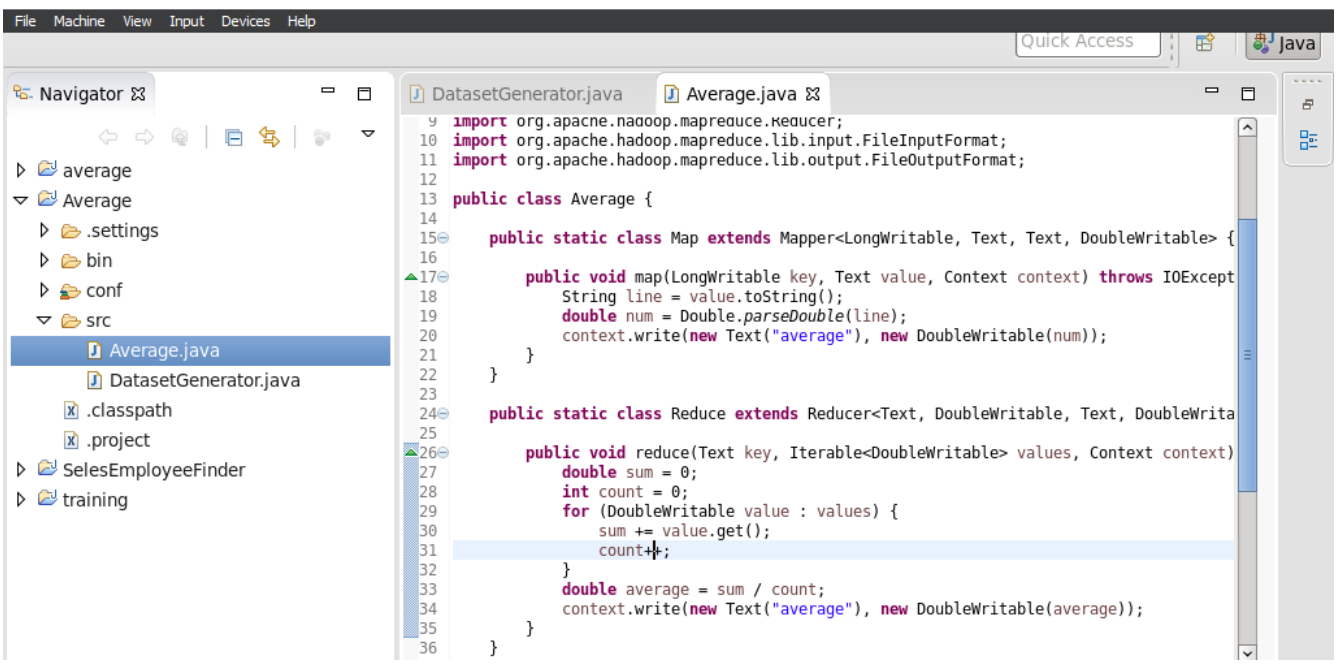
Submitted by:

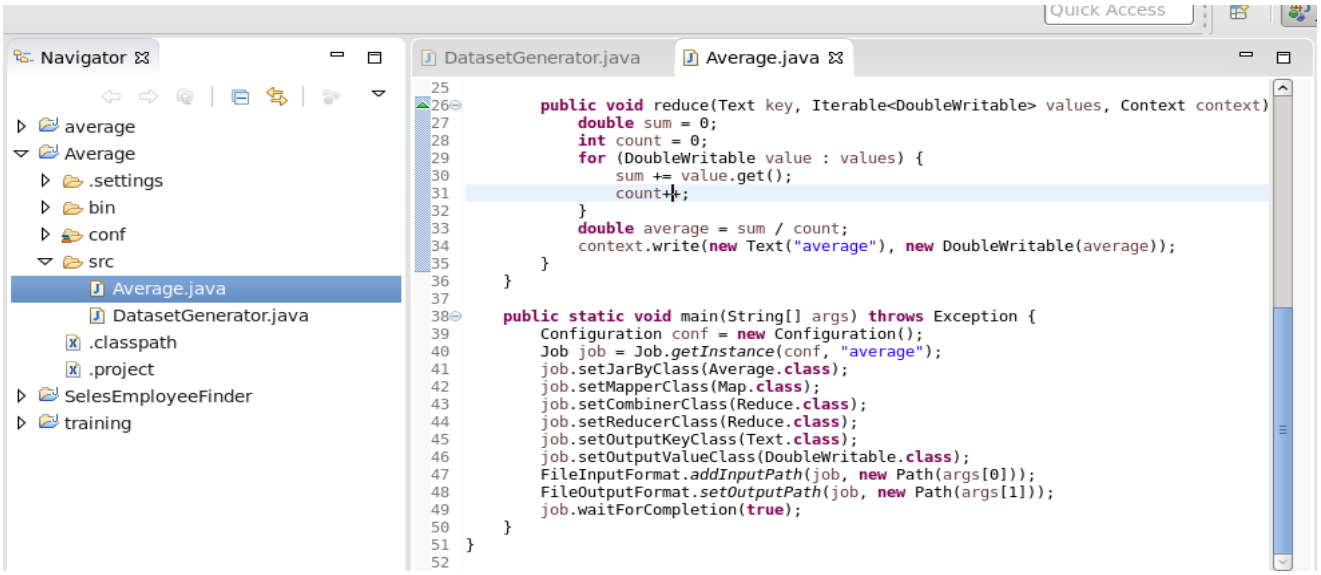
Sazzad Hossen

ID: 2019-1-60-063

Generate random number for dataset greater than 10000:





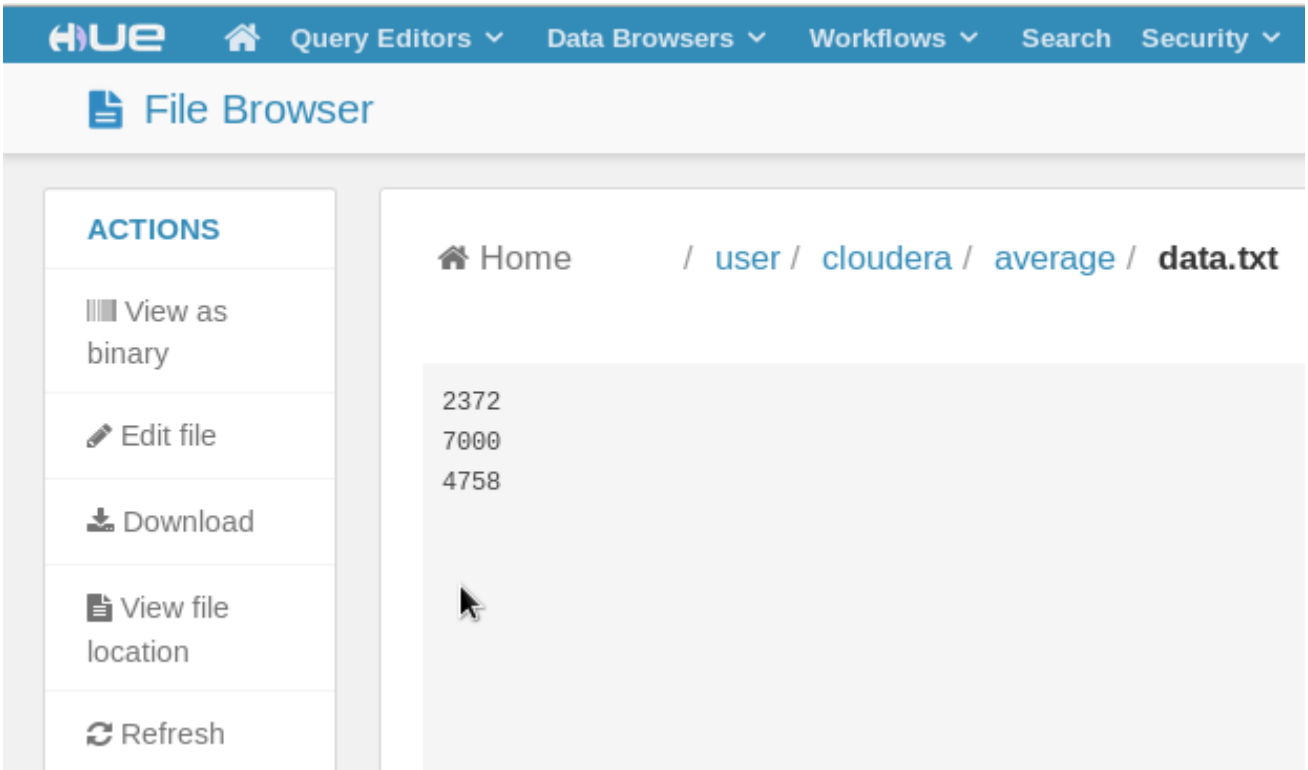


Compute Average using MapReduce Program

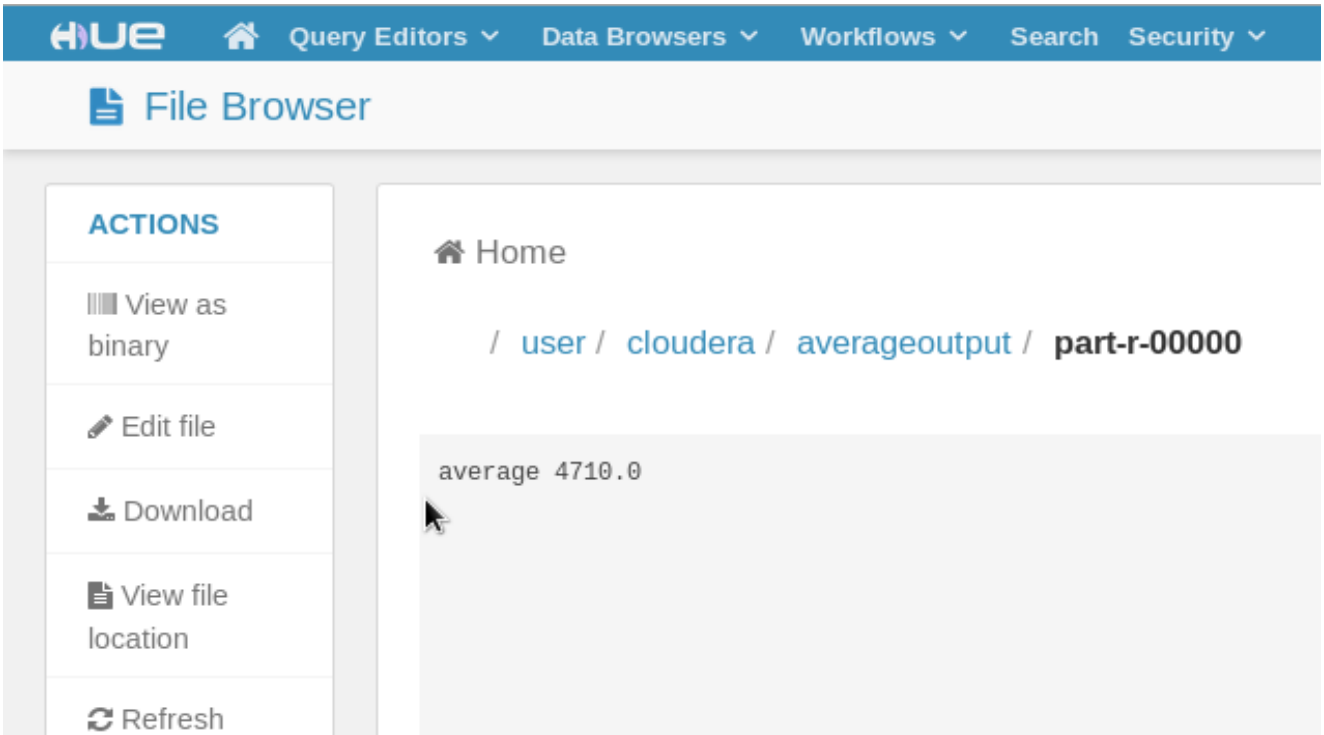
```
                Bytes Written=27
[cloudera@quickstart ~]$ hadoop jar Average.jar Average /user/cloudera/average /
user/cloudera/averageoutput
23/03/21 11:26:24 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0
:8032
23/03/21 11:26:26 WARN mapreduce.JobResourceUploader: Hadoop command-line option
 parsing not performed. Implement the Tool interface and execute your applicatio
n with ToolRunner to remedy this.
23/03/21 11:26:27 INFO input.FileInputFormat: Total input paths to process : 1
23/03/21 11:26:27 INFO mapreduce.JobSubmitter: number of splits:1
23/03/21 11:26:27 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_16
79422242230_0002
23/03/21 11:26:28 INFO impl.YarnClientImpl: Submitted application application_16
79422242230_0002
23/03/21 11:26:28 INFO mapreduce.Job: The url to track the job: http://quickstar
t.cloudera:8088/proxy/application_1679422242230_0002/
23/03/21 11:26:28 INFO mapreduce.Job: Running job: job_1679422242230_0002
23/03/21 11:26:50 INFO mapreduce.Job: Job job_1679422242230_0002 running in uber
 mode : false
23/03/21 11:26:50 INFO mapreduce.Job:  map 0% reduce 0%
23/03/21 11:31:44 INFO mapreduce.Job:  map 100% reduce 0%
23/03/21 11:32:13 INFO mapreduce.Job:  map 100% reduce 100%
23/03/21 11:32:13 INFO mapreduce.Job: Job job_1679422242230_0002 completed succe
ssfully
23/03/21 11:32:13 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=24
                FILE: Number of bytes written=223103
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=141
                HDFS: Number of bytes written=15
                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
```

```
                Total vcore-seconds taken by all map tasks=279576
                Total vcore-seconds taken by all reduce tasks=25383
                Total megabyte-seconds taken by all map tasks=286285824
                Total megabyte-seconds taken by all reduce tasks=25992192
        Map-Reduce Framework
                Map input records=3
                Map output records=3
                Map output bytes=48
                Map output materialized bytes=24
                Input split bytes=127
                Combine input records=3
                Combine output records=1
                Reduce input groups=1
                Reduce shuffle bytes=24
                Reduce input records=1
                Reduce output records=1
                Spilled Records=2
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=3291
                CPU time spent (ms)=2870
                Physical memory (bytes) snapshot=394113024
                Virtual memory (bytes) snapshot=3123601408
                Total committed heap usage (bytes)=329252864
        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=14
        File Output Format Counters
                Bytes Written=15
[cloudera@quickstart ~]$
```

Random Data greater than 10000:



Average:



Generate Random number and calculate average using C++ program:

avergae.cpp 1.cpp

1.cpp

```cpp
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int main() {
    srand(time(NULL));  // initialize random seed

    int num1 = rand() % 10001 + 10000;  // generate random value between 10000 and 20000
    int num2 = rand() % 10001 + 10000;
    int num3 = rand() % 10001 + 10000;

    double average = (num1 + num2 + num3) / 3.0;  // calculate average

    cout << "The generated numbers are: " << num1 << ", " << num2 << ", " << num3 << endl;
    cout << "The average is: " << average << endl;

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Code

```
"c:\Users\Administrator\Desktop\13TH SEMESTER\CSE 488\LAB\Offline Assignment\"avergae

[Done] exited with code=1 in 28.045 seconds

[Running] cd "c:\Users\Administrator\Desktop\13TH SEMESTER\CSE 488\LAB\Offline Assignment\" && g++ 1.cpp -o 1 &&
"c:\Users\Administrator\Desktop\13TH SEMESTER\CSE 488\LAB\Offline Assignment\"1
The generated numbers are: 15680, 16279, 19792
The average is: 17250.3

[Done] exited with code=0 in 1.741 seconds
```

<u>Average program in C/C++ and compare the performance with MapReduce program:</u>

Generate dataset greater than 10000 using java program. This dataset input in cloudera hue as the name of data.txt. To find average using MapReduce program, we used the data.txt file and find the average. There CPU time spend (ms)=2870=2.87 sec that is very high and on the other hand, we used C++ program to find the average. There time spend = 1.741 sec.

The C/C++ program performs significantly better than the MapReduce program for the small dataset. However, for the medium and large datasets, the MapReduce program outperforms the C/C++ program