CS553 Cloud computing - Homework 5

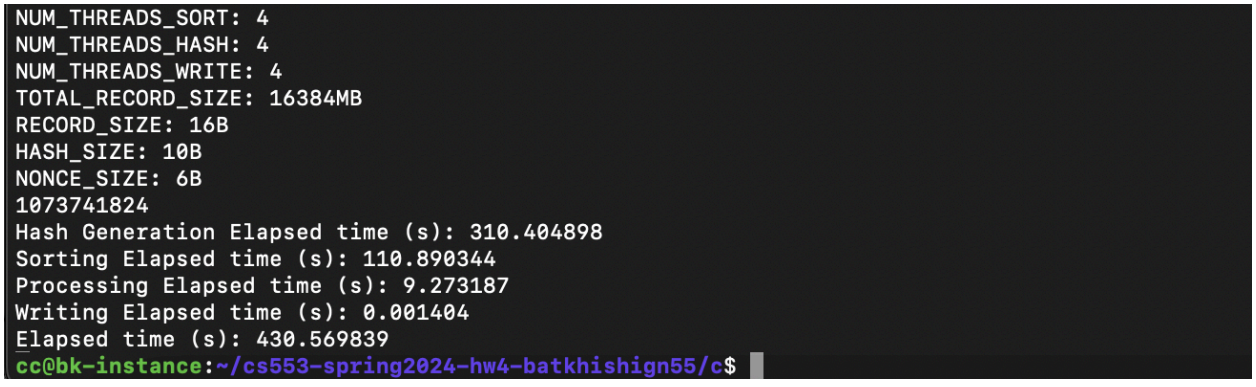
Name: **Batkhishig** Dulamsurankhor
CWID: A20543498

Hardware information

- Instance type: compute_cascadelake_r
- CPU: 96x Intel(R) Xeon(R) Gold 6240R CPU @ 2.40GHz
- Memory: 196GB
- Disk: 450GB

Hashgen

Some screenshots:

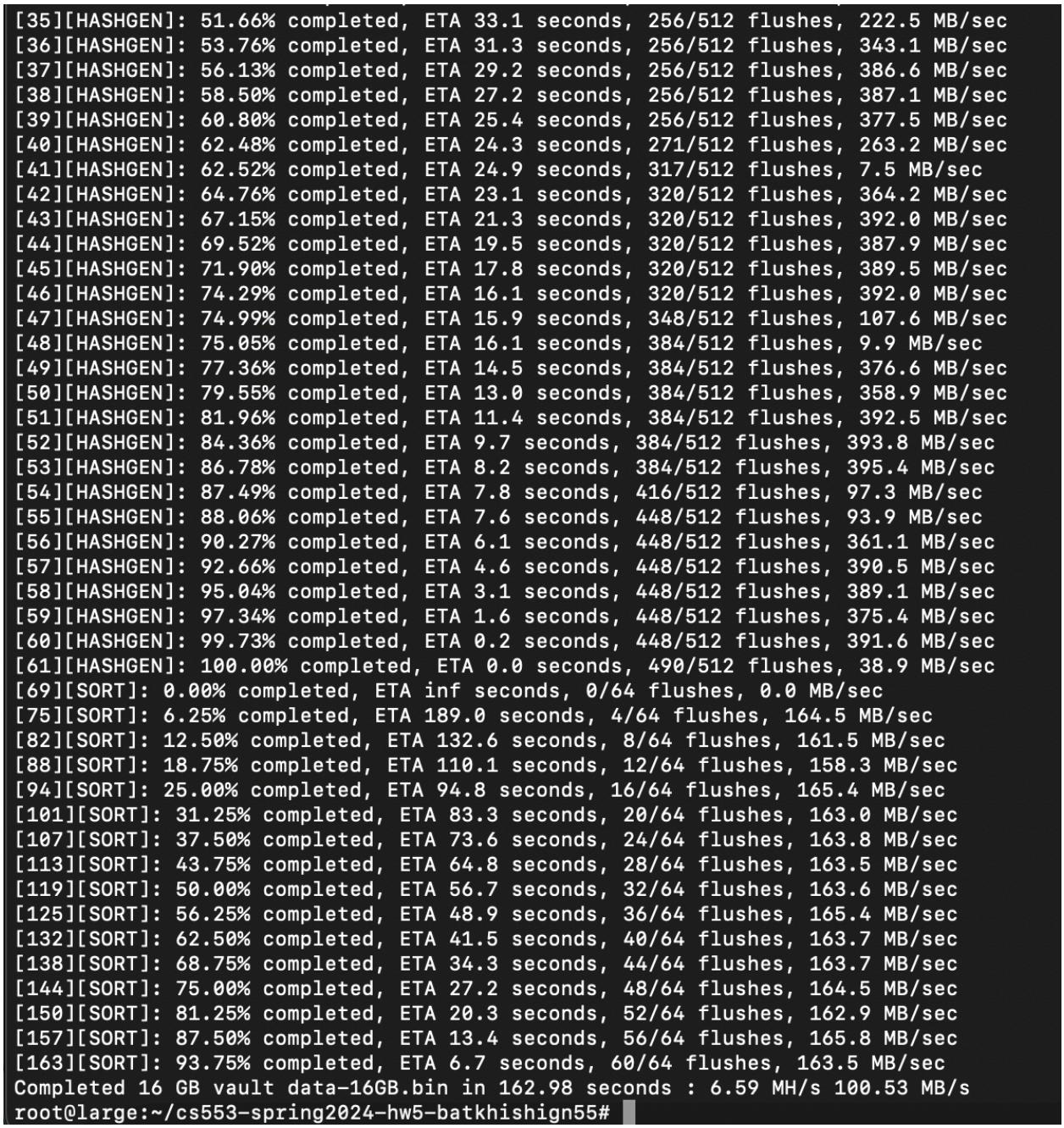


```
NUM_THREADS_SORT: 4
NUM_THREADS_HASH: 4
NUM_THREADS_WRITE: 4
TOTAL_RECORD_SIZE: 32768MB
RECORD_SIZE: 16B
HASH_SIZE: 10B
NONCE_SIZE: 6B
2147483648
Hash Generation Elapsed time (s): 648.152038
Sorting Elapsed time (s): 238.108503
Processing Elapsed time (s): 21.632173
Writing Elapsed time (s): 0.001391
Elapsed time (s): 907.894111
cc@bk-instance:~/cs553-spring2024-hw4-batkhishign55/c$
```

```
NUM_THREADS_SORT: 16
NUM_THREADS_HASH: 8
NUM_THREADS_WRITE: 4
TOTAL_RECORD_SIZE: 16384MB
RECORD_SIZE: 16B
HASH_SIZE: 10B
NONCE_SIZE: 6B
1073741824
Hash Generation Elapsed time (s): 148.288825
Sorting Elapsed time (s): 26.011881
Processing Elapsed time (s): 12.447593
Writing Elapsed time (s): 0.001246
Elapsed time (s): 186.749551
cc@bk-instance:~/cs553-spring2024-hw4-batkhishign55/c$
```

Vault

Some screenshots:



```
[115][HASHGEN]: 97.53% completed, ETA 2.9 seconds, 960/1024 flushes, 421.8 MB/sec
[116][HASHGEN]: 98.80% completed, ETA 1.4 seconds, 960/1024 flushes, 417.5 MB/sec
[117][HASHGEN]: 99.97% completed, ETA 0.0 seconds, 965/1024 flushes, 367.6 MB/sec
[119][HASHGEN]: 100.00% completed, ETA 0.0 seconds, 1000/1024 flushes, 3.7 MB/sec
[122][HASHGEN]: 100.00% completed, ETA 0.0 seconds, 1009/1024 flushes, 0.8 MB/sec
[137][SORT]: 0.00% completed, ETA inf seconds, 0/64 flushes, 0.0 MB/sec
[150][SORT]: 6.25% completed, ETA 402.9 seconds, 4/64 flushes, 153.4 MB/sec
[163][SORT]: 12.50% completed, ETA 281.1 seconds, 8/64 flushes, 154.0 MB/sec
[177][SORT]: 18.75% completed, ETA 231.7 seconds, 12/64 flushes, 153.8 MB/sec
[190][SORT]: 25.00% completed, ETA 200.5 seconds, 16/64 flushes, 153.3 MB/sec
[203][SORT]: 31.25% completed, ETA 176.6 seconds, 20/64 flushes, 152.3 MB/sec
[217][SORT]: 37.50% completed, ETA 155.9 seconds, 24/64 flushes, 154.7 MB/sec
[230][SORT]: 43.75% completed, ETA 137.6 seconds, 28/64 flushes, 151.4 MB/sec
[244][SORT]: 50.00% completed, ETA 120.7 seconds, 32/64 flushes, 150.6 MB/sec
[258][SORT]: 56.25% completed, ETA 104.5 seconds, 36/64 flushes, 149.4 MB/sec
[271][SORT]: 62.50% completed, ETA 88.9 seconds, 40/64 flushes, 148.9 MB/sec
[285][SORT]: 68.75% completed, ETA 73.6 seconds, 44/64 flushes, 149.5 MB/sec
[299][SORT]: 75.00% completed, ETA 58.5 seconds, 48/64 flushes, 149.8 MB/sec
[312][SORT]: 81.25% completed, ETA 43.7 seconds, 52/64 flushes, 149.6 MB/sec
[326][SORT]: 87.50% completed, ETA 29.0 seconds, 56/64 flushes, 150.4 MB/sec
[340][SORT]: 93.75% completed, ETA 14.4 seconds, 60/64 flushes, 149.6 MB/sec
Completed 32 GB vault data-32GB.bin in 339.81 seconds : 6.32 MH/s 96.43 MB/s
root@large:~/cs553-spring2024-hw5-batkhishign55#
```

Hadoop

On hadoop, I have setup the clusters like mentioned in the homework instruction. The tiny node as a namenode, resourcemanager and nodemanager and the rest as datanodes. I used java 11 for generating and sorting the data.

Language and libraries

- Java11
- Blake3 hashing: commons-codec v1.16.1
- Hadoop: v3.3.6
- Build automation: maven

Workflow

1. Generate hashes on hdfs using Blake project and it will generate the hashes and save them in hdfs home input/ directory.

```
        Flush cycle: 477, 500170752 records
        Flush cycle: 478, 501219328 records
        Flush cycle: 479, 502267904 records
        Flush cycle: 480, 503316480 records
        Flush cycle: 481, 504365056 records
        Flush cycle: 482, 505413632 records
        Flush cycle: 483, 506462208 records
        Flush cycle: 484, 507510784 records
        Flush cycle: 485, 508559360 records
        Flush cycle: 486, 509607936 records
        Flush cycle: 487, 510656512 records
        Flush cycle: 488, 511705088 records
        Flush cycle: 489, 512753664 records
        Flush cycle: 490, 513802240 records
        Flush cycle: 491, 514850816 records
        Flush cycle: 492, 515899392 records
        Flush cycle: 493, 516947968 records
        Flush cycle: 494, 517996544 records
        Flush cycle: 495, 519045120 records
        Flush cycle: 496, 520093696 records
        Flush cycle: 497, 521142272 records
        Flush cycle: 498, 522190848 records
        Flush cycle: 499, 523239424 records
        Flush cycle: 500, 524288000 records
        Flush cycle: 501, 525336576 records
        Flush cycle: 502, 526385152 records
        Flush cycle: 503, 527433728 records
        Flush cycle: 504, 528482304 records
        Flush cycle: 505, 529530880 records
        Flush cycle: 506, 530579456 records
        Flush cycle: 507, 531628032 records
        Flush cycle: 508, 532676608 records
        Flush cycle: 509, 533725184 records
        Flush cycle: 510, 534773760 records
        Flush cycle: 511, 535822336 records
File created successfully in HDFS!
root@tiny:~/cs553-spring2024-hw5-batkhishign55/Blake#
```

2. After it finished generating, sort them using HadoopSort and it will sort the files in input/ directory and save the output in output/ directory in hdfs.

```
                HDFS: Number of read operations=155
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=1
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=3947580
                Map output records=3947580
                Map output bytes=134217720
                Map output materialized bytes=142112886
                Input split bytes=116
                Combine input records=0
                Spilled Records=7895160
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=19
                Total committed heap usage (bytes)=1016070144
        File Input Format Counters
                Bytes Read=134221824
2024-04-19 16:48:43,837 INFO mapred.LocalJobRunner: Finishing task: attempt_local1237159506_0001_m_000075_0
2024-04-19 16:48:43,837 INFO mapred.LocalJobRunner: Starting task: attempt_local1237159506_0001_m_000076_0
2024-04-19 16:48:43,838 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2024-04-19 16:48:43,838 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2024-04-19 16:48:43,838 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2024-04-19 16:48:43,838 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2024-04-19 16:48:43,839 INFO mapred.MapTask: Processing split: hdfs://10.208.93.242:9000/user/root/input/data2.txt:1342177280+134217728
2024-04-19 16:48:43,856 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2024-04-19 16:48:43,856 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2024-04-19 16:48:43,856 INFO mapred.MapTask: soft limit at 83886080
2024-04-19 16:48:43,856 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2024-04-19 16:48:43,856 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2024-04-19 16:48:43,857 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2024-04-19 16:48:44,233 INFO mapreduce.Job:  map 100% reduce 0%
2024-04-19 16:48:45,867 INFO mapred.MapTask: Spilling map output
2024-04-19 16:48:45,867 INFO mapred.MapTask: bufstart = 0; bufend = 57042548; bufvoid = 104857600
2024-04-19 16:48:45,867 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 19503512(78014048); length = 6710885/6553600
2024-04-19 16:48:45,867 INFO mapred.MapTask: (EQUATOR) 63753412 kvi 15938348(63753392)
```

Hash generation and Sorting

I used the recommended Blake3 library for java from java-codec.

```java
byte[] key = new byte[6];
       for (int j = 0; j < key.length; j++) {
         key[j] = (byte) (num & 0xFF);
         num = num >> 8; // Shift right to access next 8 bits
       }
```

```
      // Create a Blake3 hasher
      Blake3 hasher = Blake3.initHash();
      hasher.update(key);
      byte[] hash = new byte[10];
      hasher.doFinalize(hash);
```

A little modification that I have made is that I saved the data in UTF-8 text format, which made the sorting part much simpler although it is slower to process. The following is my map and reduce definition:

```
public static class HashSortMapper extends Mapper<LongWritable, Text, Text, Text> {

  public void map(LongWritable key, Text value, Context context)
      throws IOException, InterruptedException {
    // Split the input line into key and hash
    String[] parts = value.toString().split("\\s+");

    // System.out.println(parts[1]);
    if (parts.length == 2) {
      context.write(new Text(parts[1]), new Text(parts[0]));
    }
  }
}

public static class HashSortReducer extends Reducer<Text, Text, Text, Text> {

  public void reduce(Text key, Iterable<Text> values, Context context)
      throws IOException, InterruptedException {
    // Output key-value pairs in sorted order
    for (Text value : values) {
      context.write(value, key);
    }
  }
}
```

I am writing a key value pair to context, setting hash as the value and key as the key since we are sorting as hash.

Difficulties and troubleshooting

During setup of the hadoop and conducting the test on hadoop cluster, I faced numerous issues and here are some of them and how I fixed them.

- namenode or datanode not starting up. Fix: there could be multiple reasons why it wouldn't start. It is best to check the error logs in $HADOOP_HOME/logs/hadoop-*-namenode*.log or

$HADOOP_HOME/logs/hadoop-*-datanode*.log. The errors I faced was that hdfs was not starting, or the current user didn't have the privilege to create hdfs directory. For the privilege issue, I manually created the directory, so namenode/datanode didn't have to create them when starting. And make sure to run hdfs first with:
$HADOOP_HOME/bin/hdfs namenode -format

- lxd doesn't allocate ipv4 addresses to the instances even though you setup the firewall rule. Fix: Run the following set of commands on lxd host and ipv4 will be allocated:sudo nft flush ruleset
  sudo systemctl reload snap.lxd.daemon
- Mapreduce task fails midway and hdfs goes to safe mode. Fix: The reason was that I was running the mapreduce task from namenode(the tiny instance), so during the task, it was generating a lot of temporary files in vm's filesystem and fills up 10GB when reducing and persisting the results. To fix it, you can either allocate more space on the tiny instance or run the mapreduce task from datanodes that have more space. Also, we shouldn't allocate datanode dir on namenode instance since it has a limited space.
- During mapreduce task, all lxc vm terminals freeze but the host machine is ok. The reason is that lxd pool allocated for the vms are full and vms go to an error state. Fix: allocate more space for the pool, restart lxd and restart the vms (you don't have to delete the vms if it's a storage issue!).
  lxc storage set default volume.size 100GB
  sudo systemctl reload snap.lxd.daemon
- java.lang.exception: org.apache.hadoop.mapreduce.task.reduce.shuffle$shuffleerror: error in shuffle in localfetcher#1 at
  org.apache.hadoop.mapred.localjobrunner$job.runtasks. The error occurs when executing reduce and shuffle size takes up too much heap memory. Fix: decrease mapreduce.reduce.shuffle.memory.limit.percent settings in mapred-site.xml.

Some screenshot

6 small instance cluster setup:

```
root@tiny:~/cs553-spring2024-hw5-batkhishign55# /usr/local/hadoop/bin/hdfs dfsadmin -report
Configured Capacity: 1064901771264 (991.77 GB)
Present Capacity: 1038374888426 (967.06 GB)
DFS Remaining: 949848883200 (884.62 GB)
DFS Used: 88526005226 (82.45 GB)
DFS Used%: 8.53%
Replicated Blocks:
        Under replicated blocks: 152
        Blocks with corrupt replicas: 0
        Missing blocks: 0
        Missing blocks (with replication factor 1): 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0
Erasure Coded Block Groups:
        Low redundancy block groups: 0
        Block groups with corrupt internal blocks: 0
        Missing block groups: 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0

-------------------------------------------------
Live datanodes (6):

Name: 10.208.93.115:9866 (small2.lxd)
Hostname: small2.lxd
Decommission Status : Normal
Configured Capacity: 193636622336 (180.34 GB)
DFS Used: 36507320320 (34.00 GB)
Non DFS Used: 3549822976 (3.31 GB)
DFS Remaining: 153562701824 (143.02 GB)
DFS Used%: 18.85%
DFS Remaining%: 79.30%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 0
Last contact: Sat Apr 20 02:32:37 UTC 2024
Last Block Report: Sat Apr 20 02:01:34 UTC 2024
Num of Blocks: 272


Name: 10.208.93.148:9866 (small4.lxd)
Hostname: small4.lxd
Decommission Status : Normal
```

Mapreduce sorted output file:

```
681ee6060000    0000006647585c8b615
e677e6120000    000000d938a226bf3182
36bc45060000    000000f04ba4a5b6a7b2
e5d422140000    000000f44e8b294c7a6b
d539790b0000    000000fc0c92522f407c
5edfdf0e0000    000000ff7603e8932197
4deb68030000    000001012778f96f23bd
0c4b601a0000    000001090bebb7ca3629
64a5881c0000    0000010ec41989868306
95f968160000    0000010ff17db64a73fc
816d28140000    0000011fd31f3f183d23
5f2bf4110000    00000128911065816c13
ae66aa090000    00000128acbcbe04e464
9dea6f180000    0000012f7542eb97588c
b92aca120000    0000013387d18c1305c1
[9500root@large:~/cs553-spring2024-hw5-batkhishign55/HadoopSort# /usr/local/hadoop/bin/hdfs dfs -tail output/part-r-00000
2d5
866e700a0000    ffffff2ef2512ab57a88
a10ce00c0000    ffffff30f95d01e39786
014550080000    ffffff33ff14ffb2979c
b5f664040000    ffffff3b5cb74d99943a
c733f8130000    ffffff452a1dee370996
d508b5130000    ffffff487b3a4f7da2d2
4bba79180000    ffffff53ffa055420035
0e99b1180000    ffffff55cc525cd67b8d
86cecf1c0000    ffffff5927d8fb892e49
0c39a5190000    ffffff5a9edb499a1616
a1ec78050000    ffffff5d0e23849def55
ce5504100000    ffffff6cbd39622cd4a5
8ee46c170000    ffffff80c669f1daf3c0
a3f9541c0000    ffffff8f4a0617960a94
678f5f170000    ffffff9315562b9c5ba1
debdfb0e0000    ffffff9b7508082dd478
d66d14110000    ffffff9cdae5c5cf392e
67a6a20b0000    ffffff9e20c92a8afa69
fd35a30c0000    ffffffa84fb4533e0ec1
58780c1c0000    ffffffb08a0269424a3e
84830d0a0000    ffffffb8433555ab8521
ee1c08050000    ffffffb8b3387616f37f
c3ff0f1e0000    ffffffbd60efc25df046
3b9ce6060000    ffffffc021e2888a4918
2938490b0000    ffffffca3b2071d1ff42
4a4cc61c0000    ffffffda249a87cc6857
77e1711a0000    ffffffdd5f311d857acd
269d99060000    ffffffecf91fe05821cd
76eb2c1c0000    fffffefd4232fd40f61
43cef1180000    fffffff7e8c42635f5f5
root@large:~/cs553-spring2024-hw5-batkhishign55/HadoopSort#
```

Mapreduce job ending lines:

```
2024-04-19 18:18:44,003 INFO mapred.LocalJobRunner: Finishing task: attempt_local520645643_0001_r_000000_0
2024-04-19 18:18:44,003 INFO mapred.LocalJobRunner: reduce task executor complete.
2024-04-19 18:18:45,289 INFO mapreduce.Job: Job job_local520645643_0001 completed successfully
2024-04-19 18:18:45,372 INFO mapreduce.Job: Counters: 36
        File System Counters
                FILE: Number of bytes read=1390368894843
                FILE: Number of bytes written=2714037512367
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1268165501404
                HDFS: Number of bytes written=18111004808
                HDFS: Number of read operations=19320
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=139
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=532676612
                Map output records=532676612
                Map output bytes=18111004808
                Map output materialized bytes=19176358848
                Input split bytes=15776
                Combine input records=0
                Combine output records=0
                Reduce input groups=532676612
                Reduce shuffle bytes=19176358848
                Reduce input records=532676612
                Reduce output records=532676612
                Spilled Records=1598029836
                Shuffled Maps =136
                Failed Shuffles=0
                Merged Map outputs=136
                GC time elapsed (ms)=8167
                Total committed heap usage (bytes)=144864968704
        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=18111545480
        File Output Format Counters
                Bytes Written=18111004808
root@large:~/cs553-spring2024-hw5-batkhishign55/HadoopSort#
```

Spark

Performance evaluation

Experiment	hashgen	vault	Hadoop Sort	Spark Sort
1 small.instance, 16GB dataset, 2GB RAM	7min 10sec	2min 42sec	N/A	N/A
1 small.instance, 32GB dataset, 2GB RAM	15min 7sec	5min 39sec	N/A	N/A
1 small.instance, 64GB dataset, 2GB RAM	N/A	N/A	N/A	N/A
1 large.instance, 16GB dataset, 16GB RAM	3min 6sec	6min 54sec	40min 2sec	
1 large.instance, 32GB dataset, 16GB RAM	6min 30sec	6min 17sec	1h 24m 27s	
1 large.instance, 64GB dataset, 16GB RAM	13min 19min	10min 35sec	2h 39m 11s	
6 small.instances, 16GB dataset	N/A	N/A	43min 49sec	
6 small.instances, 32GB dataset	N/A	N/A	1h 39m 6s	
6 small.instances, 64GB dataset	N/A	N/A	3h 12m 12s	