# CS 6343: CLOUD COMPUTING

## Project #1Report

### Akshay Thakare

### (netId: ast140230)

## OBJECTIVES

The purpose of this study is to observe the behavior of the HDFS Map Reduce. We are given sanitized data from <http://www.police.uk/data>. To compute the total number of crime incidents of each crime type in each region we have to write the Map Reduce program. The crime region is defined by a coordinate system (East, North). East and North are defined by 6 digit numbers.

We have considered following definitions for a region:

1. Region definition 1: Using only the first digits of the coordinate system.
2. Region definition 2: Using the first three digits of the coordinate system.
3. Region definition 3: Using the first five digits of the coordinate system.
4. There are different types of crime viz. Anti social behavior, Burglary, Drugs etc.

# Distribution of files in HDFS

**Input:**

1. A small text file:

|  |  |
| --- | --- |
| Name | Size (Kbs) |
| 2012-02-northamptonshire-street.csv | 1114764 bytes |

1. Set of small files(114 small size files): 191653604 Bytes
2. One large text file:

|  |  |
| --- | --- |
| largeFile.csv | [2183910057](tel:2183910057) bytes |

**Output:**

1. For the first case, where the input file is a single text file of size 1114764Bytes, the corresponding status of block is:

**hduser@hdfsMaster:~/MapReduceAssignment$ hadoop fsck /MapReduCeAssignmentInput/2012-02-northamptonshire-street.csv  -files -locations -blocks**

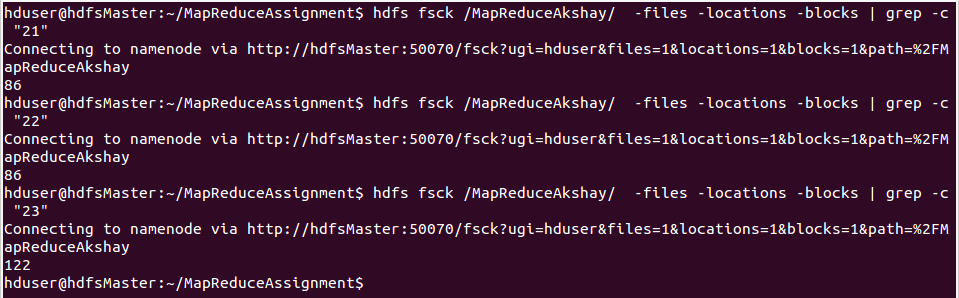
Status: HEALTHY  
  Total size:    1114764 B  
  Total dirs:    0  
  Total files:    1  
  Total symlinks:        0  
  Total blocks (validated):    1 (avg. block size 1114764 B)  
  Minimally replicated blocks:    1 (100.0 %)  
  Over-replicated blocks:    0 (0.0 %)  
  Under-replicated blocks:    0 (0.0 %)  
  Mis-replicated blocks:        0 (0.0 %)  
  Default replication factor:    2  
  Average block replication:    2.0  
  Corrupt blocks:        0  
  Missing replicas:        0 (0.0 %)  
  Number of data-nodes:        3  
  Number of racks:        1

|  |  |  |
| --- | --- | --- |
| DataNode | Name | Size(Bytes) |
| [10.176.128.2](http://10.176.128.23:50010/)1 | blk\_1073742767\_1943 | 1114764 |
| [10.176.128.23](http://10.176.128.23:50010/) | blk\_1073742767\_1943 | 1114764 |

1. For the second case where there are multiple input files, corresponding blocks are:

**hduser@hdfsMaster:~/MapReduceAssignment$ hdfs fsck /MapReduCeAkshay/ -files -locations -blocks**

Status: HEALTHY  
 Total size:    191653604 B  
 Total dirs:    1  
 Total files:    114  
 Total symlinks:        0  
 Total blocks (validated):    114 (avg. block size 1681171 B)  
 Minimally replicated blocks:    114 (100.0 %)  
 Over-replicated blocks:    0 (0.0 %)  
 Under-replicated blocks:    0 (0.0 %)  
 Mis-replicated blocks:        0 (0.0 %)  
 Default replication factor:    2  
 Average block replication:    2.0  
 Corrupt blocks:        0  
 Missing replicas:        0 (0.0 %)  
 Number of data-nodes:        3  
 Number of racks:        1

****

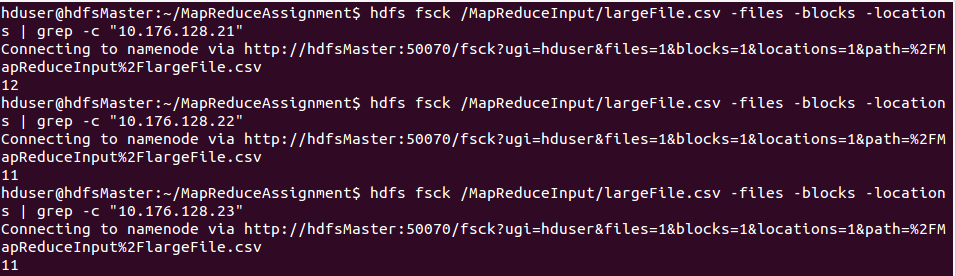
1. For the third input type (largeFile.csv )

**hduser@hdfsMaster:~/MapReduceAssignment$ hdfs fsck /MapReduCeAssignmentInput/largeFile.csv  -files -locations -blocks**

Status: HEALTHY  
 Total size:    [2183910057](tel:2183910057) B  
 Total dirs:    0  
 Total files:    1  
 Total symlinks:        0  
 Total blocks (validated):    17 (avg. block size 128465297 B)  
 Minimally replicated blocks:    17 (100.0 %)  
 Over-replicated blocks:    0 (0.0 %)  
 Under-replicated blocks:    0 (0.0 %)  
 Mis-replicated blocks:        0 (0.0 %)  
 Default replication factor:    2  
 Average block replication:    2.0  
 Corrupt blocks:        0  
 Missing replicas:        0 (0.0 %)  
 Number of data-nodes:        3  
 Number of racks:        1

|  |  |  |
| --- | --- | --- |
| DataNode | Name | Size(Bytes) |
| [10.176.128.2](http://10.176.128.23:50010/)1 | blk\_1073743509\_2694 | 134217728 |
| [10.176.128.2](http://10.176.128.23:50010/)2 | blk\_1073743509\_2694 | 134217728 |
| [10.176.128.2](http://10.176.128.23:50010/)3 | blk\_1073743510\_2695 | 134217728 |
| [10.176.128.2](http://10.176.128.23:50010/)1 | blk\_1073743510\_2695 | 134217728 |
|  | **.** |  |
|  | **.** |  |
| [10.176.128.2](http://10.176.128.23:50010/)3 | blk\_1073743525\_2710 | 36426409 |
| [10.176.128.2](http://10.176.128.23:50010/)1 | blk\_1073743525\_2710 | 36426409 |

**Distribution of blocks on all three slaves for large file:**



# MapReduce Program

## Pseudocode

*map (key, value):*

*//key: document, value: each line of the file.*

*for each easting, northing, crime type in value: emit(<easting,northing,crime type>, 1)*

*reduce (key, values):*

*//key: region definition, values: intermediate counts of the crime types.*

*totalcount = 0*

*for each v in values: totalcount += v*

*emit (key, result)*

**Run: Small input files, key = first digit of easting and northing:**

Map-Reduce Framework Values:

|  |  |
| --- | --- |
| Job ID | job\_local1372024914\_0001 |
| number of splits | 1 |
| Map input records | 6192 |
| Map output records | 6191 |
| Map output bytes | 161325 |
| Map output materialized bytes | 173713 |
| Input split bytes | 149 |
| Combine input records | 0 |
| Combine output records | 0 |
| Reduce input groups | 21 |
| Reduce shuffle bytes | 173713 |
| Reduce input records | 6191 |
| Reduce output records | 21 |
| Spilled Records | 12382 |
| Shuffled Maps | 1 |
| Failed Shuffles | 0 |
| Merged Map outputs | 1 |
| GC time elapsed (ms) | 21 |
| Total committed heap usage (bytes) | 352976896 |

File System Counters:

  FILE: Number of bytes read=16833146  
     FILE: Number of bytes written=17684969  
     FILE: Number of read operations=0  
     FILE: Number of large read operations=0  
     FILE: Number of write operations=0  
     HDFS: Number of bytes read=2229528  
     HDFS: Number of bytes written=464  
     HDFS: Number of read operations=13  
     HDFS: Number of large read operations=0  
     HDFS: Number of write operations=4

**Run: Set of small input files, key = first digit of easting and northing:**

Map-Reduce Framework Values:

|  |  |
| --- | --- |
| Job ID | job\_local1250855683\_0001 |
| number of splits | 114 |
| Map input records | 1196750 |
| Map output records | 1186496 |
| Map output bytes | 30073478 |
| Map output materialized bytes | 32447154 |
| Input split bytes | 18349 |
| Combine input records | 0 |
| Combine output records | 0 |
| Reduce input groups | 461 |
| Reduce shuffle bytes | 32447154 |
| Reduce input records | 1186496 |
| Reduce output records | 461 |
| Spilled Records | 2372992 |
| Shuffled Maps | 114 |
| Failed Shuffles | 0 |
| Merged Map outputs | 114 |
| GC time elapsed (ms) | 9146 |
| Total committed heap usage (bytes) | 20296171520 |

File System Counters:

  FILE: Number of bytes read=1041124054  
     FILE: Number of bytes written=3596359765  
     FILE: Number of read operations=0  
     FILE: Number of large read operations=0  
     FILE: Number of write operations=0  
     HDFS: Number of bytes read=15433681713  
     HDFS: Number of bytes written=10569  
     HDFS: Number of read operations=13686  
     HDFS: Number of large read operations=0  
     HDFS: Number of write operations=117

**Run: Large File, key = first digit of easting and northing:**

Map-Reduce Framework Values:

|  |  |
| --- | --- |
| Job ID | job\_local2125288115\_0001 |
| number of splits | 17 |
| Map input records | 15669890 |
| Map output records | 15565136 |
| Map output bytes | 390874802 |
| Map output materialized bytes | 422005176 |
| Input split bytes | 2159 |
| Combine input records | 0 |
| Combine output records | 0 |
| Reduce input groups | 695 |
| Reduce shuffle bytes | 422005176 |
| Reduce input records | 15565136 |
| Reduce output records | 695 |
| Spilled Records | 31130272 |
| Shuffled Maps | 17 |
| Failed Shuffles | 0 |
| Merged Map outputs | 17 |
| GC time elapsed (ms) | 2048 |
| Total committed heap usage (bytes) | [3519283200](tel:3519283200) |

File System Counters:

   FILE: Number of bytes read=992678042  
      FILE: Number of bytes written=[5199572711](tel:5199572711)  
      FILE: Number of read operations=0  
      FILE: Number of large read operations=0  
      FILE: Number of write operations=0  
      HDFS: Number of bytes read=22622119250  
      HDFS: Number of bytes written=17122  
      HDFS: Number of read operations=397  
      HDFS: Number of large read operations=0  
      HDFS: Number of write operations=20

**Observations:**

1. Map-Reducer will split the input chunks into one or more parts, depending upon the size of total input. In our experiment:-
   1. Small input file : 1 split of the input chunk
   2. Set of small files: 114 split of the total inputchunk
   3. One large file : 17 splits of the total inputchunk
2. The NameNode being a JobTracker itself, will divide the task into TaskTrackers depending of number of splits. Only one mapper would be used during execution. To achieve fault-tolerance other secondary mappers can be used. In case of failure, secondary mapper will come into the picture and will help complete the task.
3. Time taken by each mapper will depend upon the size of input given by the us. If the input size is large, every single time the buffer gets filled up and mapper has to spill the output onto the disk in order to get a free buffer.
4. Variables related to Map-Reduce framework:
   1. Reduce shuffle bytes: The shuffled map output on the reducer, which was obtained through the Task Tracker using the intermediate files of Map phase.
   2. Spilled Records: Whenever the buffer memory gets filled and there is no space left for new records, old records are written to the disk.
   3. Map output bytes: Output from the map phase of execution.
   4. Combine input records: Records merged by the combiner after the map phase.
   5. Reduce input records: Total number of values associate to the keys from the mapper phase.
   6. Reduce input groups: The number of unique keys emitted to reducer from mappers.
   7. Reduce output records: The number of records combined by all reducers.
   8. Combine output records: The number of records combined during the reduce phase.
5. **Error handling**: The execution engine stops the failed map tasks whenever there are any execution errors like unhandled exceptions.
6. **Distribution of tasks**: The map tasks are run on data nodes where the blocks are actually situated because of the data locality principle. Additional map tasks were created to handle the load.