ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

Laboratorio 5 – Hadoop Quick Start

Screenshots

Parte a)

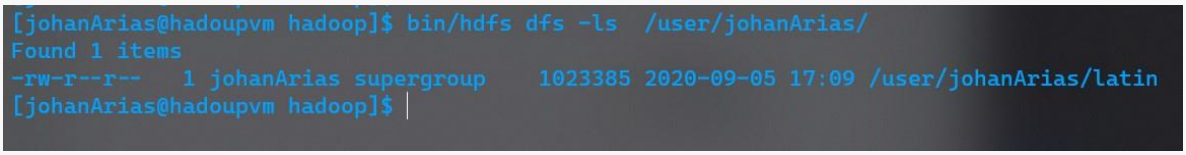


FIG 1: Archivo cargado al directorio de hdfs (bin/hdfs dfs -put latin.txt latin)

Parte b)

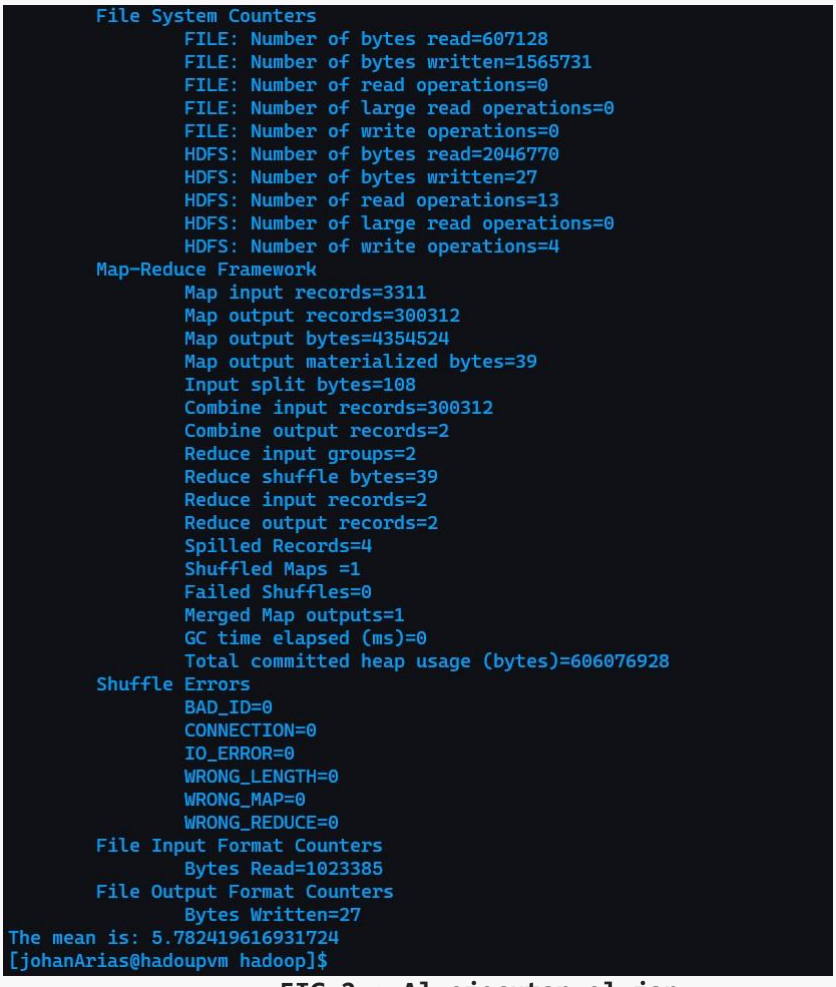


FIG 2 : Al ejecutar el jar (bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.2.jar wordmean <inputFile> <outputFile>)

```
[johanArias@hadoupvm hadoop]$ bin/hdfs dfs -cat /user/johanArias/latin_wordmean_output/*
count   150156
length  868265
[johanArias@hadoupvm hadoop]$
```

FIG 3 : Conteo de palabras y longitud calculadas exitosamente

Parte c)

```
[johanArias@hadoupvm hadoop]$ bin/hdfs dfs -ls  /user/johanArias/
Found 3 items
-rw-r--r--   1 johanArias supergroup    1980034 2020-09-05 17:35 /user/johanArias/MyWordList
-rw-r--r--   1 johanArias supergroup    1023385 2020-09-05 17:09 /user/johanArias/latin
drwxr-xr-x   - johanArias supergroup          0 2020-09-05 17:18 /user/johanArias/latin_wordmean_output
[johanArias@hadoupvm hadoop]$
```

FIG 4: Cargando mi propio archivo de palabras a hdfs
(bin/hdfs dfs -put
directory-list-2.3-medium.txt MyWordlist)

```
20/09/05 17:51:05 INFO mapreduce.Job: Counters: 35
        File System Counters
                FILE: Number of bytes read=607138
                FILE: Number of bytes written=1561265
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=3960068
                HDFS: Number of bytes written=28
                HDFS: Number of read operations=13
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Map-Reduce Framework
                Map input records=220560
                Map output records=452108
                Map output bytes=6555566
                Map output materialized bytes=39
                Input split bytes=113
                Combine input records=452108
                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)=5
                Total committed heap usage (bytes)=605028352
        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=1980034
        File Output Format Counters
                Bytes Written=28
The mean is: 7.75903545170623
[johanArias@hadoupvm hadoop]$
```

FIG 5: Calculando la media con mi propia lista de palabras
(bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.2.jar
wordmean MyWordList MyWordListOutPut)

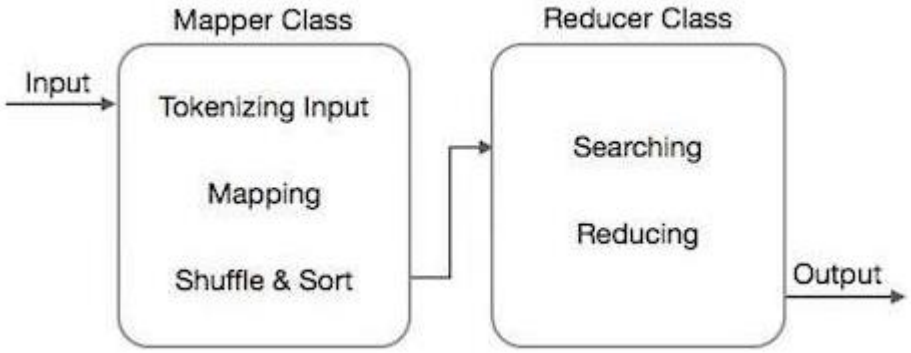
```
[johanArias@hadoupvm hadoop]$ bin/hdfs dfs -cat /user/johanArias/MyWordListOutPut/*
count   226054
length  1753961
[johanArias@hadoupvm hadoop]$
```

FIG 6: Verificando que el conteo y la longitud se hayan calculado correctamente
(bin/hdfs dfs -cat /user/johanArias/MyWordListOutPut/*)

Parte d) Investigación

El algoritmo MapReduce contiene dos tareas importantes, Mapear y Reducir. La tarea de mapear es realizada por medio de la clase "Mapper" y la tarea de reducir la realiza la clase "Reducer"

La clase Mapper toma la información, la muestra, la mapea y la clasifica. La salida de la clase Mapper se utiliza como entrada por la clase Reducer, que a su vez busca los pares coincidentes y los reduce.



MapReduce implementa varios algoritmos matemáticos para dividir una tarea en pequeñas partes y asignarlas a múltiples sistemas. En términos técnicos, el algoritmo MapReduce ayuda a enviar las tareas de Map & Reduce a los servidores apropiados en un clúster.

El algoritmo de MapReduce tiene 3 fases:

1. Map Function.
2. Suffle Function
3. Reduce Function

1. Map Function

Este es el primer paso del Algoritmo MapReduce. Toma los conjuntos de datos y los distribuye en subtareas más pequeñas. Esto se hace más adelante en dos pasos, la división y el mapeo. La división toma el conjunto de datos de entrada y divide el conjunto de datos mientras que el mapeo toma esos subconjuntos de datos y realiza la acción requerida. La salida de esta función es un par clave-valor.

2. Shuffle Function

Esto también se conoce como función de combinación e incluye la fusión y la clasificación. La fusión combina todos los pares clave-valor. Todos ellos tendrán las mismas claves. La clasificación toma la información del paso de fusión y ordena todos los pares clave-valor haciendo uso de las claves. Este paso también regresará a los pares clave-valor. La salida será ordenada.

3.Reduce Function

Este es el último paso de este algoritmo. Toma los pares clave-valor del shuffle y reduce la operación.