Semana 2 - Introducción Hadoop

Dia 9/6/2021

Ejericicos realizados por Paula Iglesias (comandos en color naranja ejecutados)

/cd1_Descubriendo HDFS

Para listar archivos: hdfs dfs -ls

1. Lo primero que tenemos que hacer es habilitar la transferencia de ficheros entre vuestra máquina virtual y vuestro host. En las settings del VMWare asociadas a la VMCloudera que tenéis, habilitad la opción shared folders y añadid una carpeta en vuestros host Windows desde donde copiaremos los archivos necesarios para trabajar.
2. Una vez creada, arrancad la MV. La carpeta en Windows compartida estará en la ruta "/mnt/hgfs"
3. Abre un terminal en la MV y cámbiale el idioma a español
4. Escribe el siguiente comando
    a. `Hadoop fs`
    b. Verás un mensaje describiendo todos los comandos posibles que contiene la FsShell
5. Escribe
    a. `Hadoop fs -ls /`
    b. Esto muestra el contenido del directorio root en HDFS // **hdfs dfs -ls /**
6. Crea un directorio local de trabajohadooop
    a. Por ejemplo /home/cloudera/ejercicios
    `mkdir ejercicios`
    `rmdir ejercios (si está vacio)`
    `rm -r → borrar en cascada`
    `cp origen destino`

    b. Copia el archivo "`shakespeare.tar.gz`" en ese directorio
    `cp origen destino`
    c. Descomprime el archivo cd ..
        i. `"tar zxvf shakespeare.tar.gz"`
    d. Copia la carpeta que acabas de descomprimir en HDFS en la ruta /user/cloudera/shakespeare. Tu home a partir de ahora será /user/cloudera
        i. `"hadoop fs -put shakespeare /user/cloudera/shakespeare"`
        ii. **hdfs dfs -put shakespeare /user/cloudera/shakespeare**
7. Lista el contenido de tu home en HDFS para comprobar que se ha copiado la carpeta shakespeare.
        i. `"hadoop fs -ls /user/cloudera"`
8. Observa el contenido de la carpeta shakespeare en hdfs
        i. `"hadoop fs -ls /user/cloudera/shakespeare"`
        ii. **hdfs dfs -ls -C /user/cloudera/shakespeare // Listar en formato lista**

9. Borra la subcarpeta “glossary” de la carpeta shakespeare en HDFS
    a. “`hadoop fs -rm /user/cloudera/shakespeare/glossary`”

10. Comprueba que se ha borrado
11. Lista las primeras 50 últimas líneas de la subcarpeta “histories”. Puedes usar los comando “cat” y “tail” (head -> primeras)
    a. “`hadoop fs -cat /user/cloudera/shakespeare/histories | tail -n 50`”
    b. hdfs dfs -cat /user/cloudera/shakespeare/histories | tail -n 50

12. Copia al Sistema de ficheros local de tu MV el fichero “poems” en la ruta /home/cloudera/ejercicios/shakespeare/shakepoems.txtop
    a. “`hadoop fs -get /user/cloudera/shakespeare/poems /home/cloudera/ejercicios/shakespeare/shakepoems.txt`”
    b. hdfs dfs -get /user/cloudera/shakespeare/poems /home/cloudera/ejercicios/shakespeare/shakepoems.txt

13. Muestra las últimas líneas de shakepoems.txt copiado en tu local por pantalla
    `tail 5 shakepoems.txt`
14. Si has terminado muy rápido, juega un poco con los comandos disponibles en la Shell de hadoop fs. Para ello introduce “hadoop fs” y observa las posibilidades.

Opciones de hadoop fs:

```
[-appendToFile <localsrc> ... <dst>]
[-cat [-ignoreCrc] <src> ...]
[-checksum <src> ...]
[-chgrp [-R] GROUP PATH...]
[-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
[-chown [-R] [OWNER][:[GROUP]] PATH...]
[-copyFromLocal [-f] [-p] [-l] <localsrc> ... <dst>]
[-copyToLocal [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
[-count [-q] [-h] [-v] [-x] <path> ...]
[-cp [-f] [-p | -p[topax]] <src> ... <dst>]
[-createSnapshot <snapshotDir> [<snapshotName>]]
[-deleteSnapshot <snapshotDir> <snapshotName>]
[-df [-h] [<path> ...]]
[-du [-s] [-h] [-x] <path> ...]
[-expunge]
[-find <path> ... <expression> ...]
[-get [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
[-getfacl [-R] <path>]
[-getfattr [-R] {-n name | -d} [-e en] <path>]
[-getmerge [-nl] <src> <localdst>]
[-help [cmd ...]]
[-ls [-C] [-d] [-h] [-q] [-R] [-t] [-S] [-r] [-u] [<path> ...]]
[-mkdir [-p] <path> ...]
[-moveFromLocal <localsrc> ... <dst>]
[-moveToLocal <src> <localdst>]
[-mv <src> ... <dst>]
[-put [-f] [-p] [-l] <localsrc> ... <dst>]
[-renameSnapshot <snapshotDir> <oldName> <newName>]
[-rm [-f] [-r|-R] [-skipTrash] <src> ...]
[-rmdir [--ignore-fail-on-non-empty] <dir> ...]
[-setfacl [-R] [{-b|-k} {-m|-x <acl_spec>} <path>]|[--set <acl_spec> <path>]]
[-setfattr {-n name [-v value] | -x name} <path>]
[-setrep [-R] [-w] <rep> <path> ...]
[-stat [format] <path> ...]
[-tail [-f] <file>]
[-test -[defsz] <path>]
[-text [-ignoreCrc] <src> ...]
[-touchz <path> ...]
[-usage [cmd ...]]
```

2_Ejecutando un MapReduce: wordcount

En este ejercicio simplemente ejecutaremos un Job consistente en la ejecución del wordcount en MapReduce sobre el dataset shakespeare. Por simplicidad, los ficheros .class y el jar ya están creados. WordCount

Como hemos comentado, wordcount cuenta el número de palabras distintas que hay en un texto dado.

Pasos a ejecutar

1. Copiar en la ruta "/home/cloudera/ejercicios" la carpeta "wordcount" y su contenido.

cp -R /mnt/Shared/wordcount /home/cloudera/ejercicios

2. Comprobar que se han copiado correctamente
3. Examinar el contenido de los tres ficheros java para asegurarnos de que están correctos.
    a. Prestar atención los parámetros de entrada de cada clase, los tipos de datos de entrada, salida e intermedios, etc.
4. La carpeta wordcount, como hemos visto, ya contiene los javas compilados y el jar creado, por lo que solo tenemos que ejecutar el submit del job hadoop usando nuestro fichero JAR para contar las ocurrencias de palabras contenidas en nuestra carpeta "shakespeare". Nuestro jar contiene las clases java compiladas dentro de un paquete llamado "solutions", por eso se le llama de este modo.hadoop
    a. "hadoop jar wordcount/wc.jar solution.WordCount

shakespeare /user/cloudera/wordcounts”

Resultado de ese comando:

```
21/06/09 06:38:05 INFO mapreduce.Job: Running job: job_1623228163079_0003
21/06/09 06:38:13 INFO mapreduce.Job: Job job_1623228163079_0003 running in uber mode : false
21/06/09 06:38:13 INFO mapreduce.Job:  map 0% reduce 0%
21/06/09 06:38:33 INFO mapreduce.Job:  map 25% reduce 0%
21/06/09 06:38:36 INFO mapreduce.Job:  map 50% reduce 0%
21/06/09 06:38:37 INFO mapreduce.Job:  map 75% reduce 0%
21/06/09 06:38:38 INFO mapreduce.Job:  map 100% reduce 0%
21/06/09 06:38:43 INFO mapreduce.Job:  map 100% reduce 100%
21/06/09 06:38:44 INFO mapreduce.Job: Job job_1623228163079_0003 completed successfully
21/06/09 06:38:44 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=10713042
                FILE: Number of bytes written=22142092
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=5284754
                HDFS: Number of bytes written=299379
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Killed map tasks=1
                Launched map tasks=4
                Launched reduce tasks=1
                Data-local map tasks=4
                Total time spent by all maps in occupied slots (ms)=75684
                Total time spent by all reduces in occupied slots (ms)=6698
                Total time spent by all map tasks (ms)=75684
                Total time spent by all reduce tasks (ms)=6698
                Total vcore-milliseconds taken by all map tasks=75684
                Total vcore-milliseconds taken by all reduce tasks=6698
                Total megabyte-milliseconds taken by all map tasks=77500416
                Total megabyte-milliseconds taken by all reduce tasks=6858752
        Map-Reduce Framework
                Map input records=173126
                Map output records=964453
                Map output bytes=8784130
                Map output materialized bytes=10713060
                Input split bytes=523
                Combine input records=0
                Combine output records=0
                Reduce input groups=29183
                Reduce shuffle bytes=10713060
                Reduce input records=964453
                Reduce output records=29183
                Spilled Records=1928906
                Shuffled Maps =4
                Failed Shuffles=0
                Merged Map outputs=4
                GC time elapsed (ms)=1199
                CPU time spent (ms)=7580
                Physical memory (bytes) snapshot=1013186560
                Virtual memory (bytes) snapshot=7527845888
                Total committed heap usage (bytes)=723206144
        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=5284231
        File Output Format Counters
                Bytes Written=299379
```

5. Una vez ejecutado, probamos a ejecutarlo nuevamente
    a. ¿Qué ocurre?
        Ya existe
6. Comprobamos el resultado de nuestro MapReduce
    a. “hadoop fs -ls /user/cloudera/wordcounts”

Resultado de ese comando:

```
[cloudera@quickstart ejercicios]$ hadoop fs -ls /user/cloudera/wordcounts
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2021-06-09 06:38 /user/cloudera/wordcounts/_SUCCESS
-rw-r--r--   1 cloudera cloudera     299379 2021-06-09 06:38 /user/cloudera/wordcounts/part-r-00000
```

7. Como solo hemos usado un reduce, vemos que solo hay un archivo de salida
   a. "/user/cloudera/wordcounts/part-r-00000"
8. Observamos el contenido del fichero
   a. "hadoop fs -cat /user/cloudera/wordcounts/part-r-00000 | less"

Resultado de ese comando:

```
AARON   72
ABERGAVENNY     9
ABHORSON        18
ABOUT   18
ABRAHAM 7
ACHILLES        88
ACT     758
ADAM    16
ADO     18
ADONIS  1
ADRIAN  13
ADRIANA 85
ADRIANO 111
ADVENTURER      1
AEGEON  20
AEMELIA 16
AEMILIA 3
AEMILIUS        11
AENEAS  58
AEacida 1
AEacides        1
AEdile  19
AEdiles 5
AEgeon  7
AEgle   1
```

   b. Escribiendo la letra "q" salimos del comando less
9. Volvemos a ejecutar el job de nuevo
   a. "hadoop jar wc.jar solution.WordCount shakespeare/poems /user/cloudera/pwords"

Resultado de ese comando:

```
21/06/09 06:41:35 INFO mapreduce.Job: Running job: job_1623228163079_0004
21/06/09 06:41:42 INFO mapreduce.Job: Job job_1623228163079_0004 running in uber mode : false
21/06/09 06:41:42 INFO mapreduce.Job:  map 0% reduce 0%
21/06/09 06:41:47 INFO mapreduce.Job:  map 100% reduce 0%
21/06/09 06:41:54 INFO mapreduce.Job:  map 100% reduce 100%
21/06/09 06:41:55 INFO mapreduce.Job: Job job_1623228163079_0004 completed successfully
21/06/09 06:41:55 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=558628
                FILE: Number of bytes written=1403623
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=268268
                HDFS: Number of bytes written=67271
                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)=3666
                Total time spent by all reduces in occupied slots (ms)=4045
                Total time spent by all map tasks (ms)=3666
                Total time spent by all reduce tasks (ms)=4045
                Total vcore-milliseconds taken by all map tasks=3666
                Total vcore-milliseconds taken by all reduce tasks=4045
                Total megabyte-milliseconds taken by all map tasks=3753984
                Total megabyte-milliseconds taken by all reduce tasks=4142080
        Map-Reduce Framework
                Map input records=7308
                Map output records=50212
                Map output bytes=458198
                Map output materialized bytes=558628
                Input split bytes=128
                Combine input records=0
                Combine output records=0
                Reduce input groups=7193
                Reduce shuffle bytes=558628
                Reduce input records=50212
                Reduce output records=7193
                Spilled Records=100424
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=103
                CPU time spent (ms)=1930
                Physical memory (bytes) snapshot=361164800
                Virtual memory (bytes) snapshot=3015159808
                Total committed heap usage (bytes)=226365440
        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=268140
        File Output Format Counters
                Bytes Written=67271
```

10. Borramos la salida producida por nuestros jobs
    a. “hadoop fs -rm -r /user/cloudera/wordcounts /user/cloudera/pwords”

Resultado de ese comando:

```
[cloudera@quickstart wordcount]$ hadoop fs -rm -r /user/cloudera/wordcounts /user/cloudera/pwords
Deleted /user/cloudera/wordcounts
Deleted /user/cloudera/pwords
```

11. Ejecutamos nuevamente nuestro job
    a. “hadoop jar wc.jar solution.WordCount shakespeare /user/cloudera/count2”
12. Mientras se ejecuta, en otro terminal ejecutamos lo siguiente, para ver la lista de Jobs que se están ejecutando
    a. “mapred job -list”

Resultado de ese comando:

```
[cloudera@quickstart ejercicios]$ mapred job -list
21/06/09 06:43:57 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
Total jobs:1
                  JobId      State           StartTime      UserName           Queue      Priority       UsedC
tainers  RsvdContainers  UsedMem         RsvdMem         NeededMem         AM info
 job_1623228163079_0005    RUNNING       1623246208754      cloudera    root.cloudera       NORMAL
      2              0    3072M              0M             3072M      http://quickstart.cloudera:8088/proxy/
plication_1623228163079_0005/
```

13. Si conocemos la id de un job, lo podemos matar. Recordemos que cerrando un terminal no se mata el job. Para ello, ejecutamos en otra terminal lo siguiente
    a. "`mapred job -kill` *`jobid`*"

Resultado de ese comando:

```
[cloudera@quickstart wordcount]$ mapred job -kill job_1623228163079_0005
21/06/09 06:44:51 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/06/09 06:44:52 INFO mapred.ClientServiceDelegate: Application state is completed. FinalApplicationStatus=SUCCEEDE
. Redirecting to job history server
21/06/09 06:44:53 INFO impl.YarnClientImpl: Killed application application_1623228163079_0005
Killed job job_1623228163079_0005
```

14. Si no te ha dado tiempo, prueba a ejecutar el job otra vez cambiándolo de nombre y prueba nuevamente.