

TECNOLOGÍAS DE COMPUTACIÓN PARA DATOS MASIVOS

Entrega 2

ANDRÉS CAMPOS CUIÑA

FECHA DE ENTREGA: 31/10/2021

ÍNDICE

1	Probar el comando <i>hdfs dfsadmin</i>	1
2	Probar el comando <i>hdfs fsck</i>	3
3	Probar el uso de códigos de borrado (<i>Erasure Codes</i>)	10

1 PROBAR EL COMANDO *HDFS DFSADMIN*

En este primer apartado lo que haremos es, en el NameNode y como usuario **hdadmin**, creamos un directorio en el directorio home del usuario **hdadmin**, al que llamaremos **auxiliar**, mediante el comando **hdfs dfs -mkdir ./auxiliar**. Una vez creado este directorio le pondremos una cuota de sólo 4 ficheros mediante el comando **hdfs dfsadmin -setQuota 4 ./auxiliar**.

Ahora lo que haremos es copiar una serie de ficheros de local a esta carpeta del HDFS. Lo primero que haremos es crear un directorio temporal, al que llamaremos **temp**, en el directorio **/opt/bd/** mediante el comando **mkdir /opt/bd/temp** y dentro de este nuevo directorio crearemos cinco ficheros vacíos mediante el comando **touch fichero{1..5}**.

Una vez creados estos ficheros comprobaremos cuántos ficheros podemos copiar al directorio **auxiliar** que creamos anteriormente en el HDFS. Para esto usaremos el comando **hdfs dfs -put /opt/temp/fichero{1..5} ./auxiliar**. Mediante este comando intentaremos copiar 5 ficheros en el directorio en el que establecimos la cuota de 4 ficheros. El resultado de la ejecución de este comando es el siguiente:

```

hdadmin@namenode: ~/temp
Archivo  Acciones  Editar  Vista  Ayuda
hdadmin@namenode: ~/temp  tcdm@tcdm-docker: ~
hdadmin@namenode:~/temp$ hdfs dfs -put /opt/bd/temp/fichero{1..5} ./auxiliar
put: The NameSpace quota (directories and files) of directory /user/hdadmin/auxiliar is exceeded:
quota=4 file count=5
put: The NameSpace quota (directories and files) of directory /user/hdadmin/auxiliar is exceeded:
quota=4 file count=5
hdadmin@namenode:~/temp$

```

Como podemos observar nos da dos errores, un error para cada uno de los ficheros **fichero4** y **fichero5**, pero copia los otros tres ficheros, como se puede ver si accedemos al *filesystem* a través de la interfaz web:

Browse Directory

Show entries

Search:

<input type="checkbox"/>	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
<input type="checkbox"/>	-rw-r--r--	hdadmin	supergroup	0 B	Oct 21 17:00	3	64 MB	fichero1	
<input type="checkbox"/>	-rw-r--r--	hdadmin	supergroup	0 B	Oct 21 17:00	3	64 MB	fichero2	
<input type="checkbox"/>	-rw-r--r--	hdadmin	supergroup	0 B	Oct 21 17:00	3	64 MB	fichero3	

Showing 1 to 3 of 3 entries

Previous
1
Next

Hadoop, 2021.

Consultando en la documentación del comando `hdfs dfsadmin -setQuota` encontramos que el propio directorio cuenta para su cuota, por lo que si queremos que se puedan guardar en un directorio un máximo de 4 ficheros, en realidad deberíamos ejecutar el comando `hdfs dfsadmin -setQuota 5 <path_to_folder>`, de esta manera se podrán almacenar 4 ficheros a mayores del propio directorio.

Finalmente, eliminamos tanto de local como del HDFS estos directorios creados de forma temporal.

2 PROBAR EL COMANDO *HDFS FSCK*

En el NameNode y como usuario hdadmin hacemos un chequeo de todo el HDFS mediante el comando `hdfs fsck /`. El resultado de la ejecución de este comando es el siguiente:

```

hdadmin@namenode: ~
Archivo Acciones Editar Vista Ayuda
hdadmin@namenode: ~ tcdm@tcdm-docker: ~
hdadmin@namenode:~$ hdfs fsck /
Connecting to namenode via http://namenode:9870/fsck?ugi=hdadmin&path=%2F
FSCK started by hdadmin (auth:SIMPLE) from /172.18.0.2 for path / at Thu Oct 21 17:17:48 CEST 202
1

/tmp/hadoop-yarn/staging/hdadmin/.staging/job_1634164496469_0001/job.jar: Under replicated BP-44
1966921-172.18.0.2-1632828351540:blk_1073741928_1104. Target Replicas is 10 but found 5 live repl
ica(s), 0 decommissioned replica(s), 0 decommissioning replica(s).

/tmp/hadoop-yarn/staging/hdadmin/.staging/job_1634164496469_0001/job.split: Under replicated BP-
441966921-172.18.0.2-1632828351540:blk_1073741929_1105. Target Replicas is 10 but found 5 live re
plica(s), 0 decommissioned replica(s), 0 decommissioning replica(s).

/tmp/hadoop-yarn/staging/hdadmin/.staging/job_1634164496469_0002/job.jar: Under replicated BP-44
1966921-172.18.0.2-1632828351540:blk_1073741949_1125. Target Replicas is 10 but found 5 live repl
ica(s), 0 decommissioned replica(s), 0 decommissioning replica(s).

/tmp/hadoop-yarn/staging/hdadmin/.staging/job_1634164496469_0002/job.split: Under replicated BP-
441966921-172.18.0.2-1632828351540:blk_1073741950_1126. Target Replicas is 10 but found 5 live re
plica(s), 0 decommissioned replica(s), 0 decommissioning replica(s).

Status: HEALTHY
Number of data-nodes: 5
Number of racks: 4
Total dirs: 27
Total symlinks: 0

Replicated Blocks:
Total size: 339369946 B
Total files: 74
Total blocks (validated): 76 (avg. block size 4465394 B)
Minimally replicated blocks: 76 (100.0 %)
Over-replicated blocks: 0 (0.0 %)
Under-replicated blocks: 4 (5.263158 %)
Mis-replicated blocks: 0 (0.0 %)

```

```

hdadmin@namenode: ~
Archivo Acciones Editar Vista Ayuda

hdadmin@namenode: ~ tcdm@tcdm-docker: ~

Replicated Blocks:
Total size: 339369946 B
Total files: 74
Total blocks (validated): 76 (avg. block size 4465394 B)
Minimally replicated blocks: 76 (100.0 %)
Over-replicated blocks: 0 (0.0 %)
Under-replicated blocks: 4 (5.263158 %)
Mis-replicated blocks: 0 (0.0 %)
Default replication factor: 3
Average block replication: 3.1052632
Missing blocks: 0
Corrupt blocks: 0
Missing replicas: 20 (7.8125 %)
Blocks queued for replication: 0

Erasure Coded Block Groups:
Total size: 0 B
Total files: 0
Total block groups (validated): 0
Minimally erasure-coded block groups: 0
Over-erasure-coded block groups: 0
Under-erasure-coded block groups: 0
Unsatisfactory placement block groups: 0
Average block group size: 0.0
Missing block groups: 0
Corrupt block groups: 0
Missing internal blocks: 0
Blocks queued for replication: 0
FSCK ended at Thu Oct 21 17:17:48 CEST 2021 in 86 milliseconds

The filesystem under path '/' is HEALTHY
hdadmin@namenode:~$

```

Como se puede ver hay 4 ficheros para los cuales el número de réplicas de sus bloques es inferior al que debería ser. Esto lo podemos ver en el mensaje de error que da: **Under replicated block. Target Replicas is 10 but found 5 live replica(s)**.

Como se puede observar, el target del número de réplicas para estos bloques es de 10 pero se han encontrado sólo cinco réplicas, por lo que nos da el error de que están *under-replicated*. Sin embargo, como podemos ver el número de réplicas por defecto en el sistema es de tres, por lo que estos bloques en realidad tienen más réplicas que el número de réplicas por defecto del sistema, sólo que estos ficheros tienen un número de réplicas establecido mayor (de 10).

Para arreglar este problema podríamos forzar a estos bloques a adaptarse al factor de replicación por defecto del resto del clúster mediante el comando `hdfs dfs -setrep -w 3 <path_to_file>`.

A continuación, paramos los datanodes de forma brusca ejecutando el comando `docker container stop datanode1 datanode2 datanode4` para que sólo queden dos datanodes activos en dos racks diferentes. Tras esperar 10 minutos, mediante el comando `hdfs dfsadmin -report` podemos comprobar que sólo quedan dos datanodes vivos:

```

hdadmin@namenode: ~
Archivo Acciones Editar Vista Ayuda
tcdm@tcdm-docker: ~ hdadmin@namenode: ~
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 Oct 30 18:33:59 CEST 2021
Last Block Report: Sat Oct 30 18:14:17 CEST 2021
Num of Blocks: 43

Dead datanodes (3):

Name: 172.18.0.3:9866 (172.18.0.3)
Hostname: datanode1
Decommission Status : Normal
Configured Capacity: 31569420288 (29.40 GB)
DFS Used: 67923968 (64.78 MB)
Non DFS Used: 11337674752 (10.56 GB)
DFS Remaining: 18536587264 (17.26 GB)
DFS Used%: 0.22%
DFS Remaining%: 58.72%
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 Oct 30 18:19:05 CEST 2021
Last Block Report: Sat Oct 30 18:14:17 CEST 2021
Num of Blocks: 30

Name: 172.18.0.4:9866 (172.18.0.4)
Hostname: datanode2
Rack: /rack1

```

Como se puede ver en la salida del comando anterior, hay 3 datanodes muertos que se corresponden con los que paramos de forma deliberada.

Volvemos a ejecutar el comando `hdfs fsck /`, el resultado de este comando esta vez es el siguiente:

```

hdadmin@namenode: ~
Archivo Acciones Editar Vista Ayuda
tcdm@tcdm-docker: ~ hdadmin@namenode: ~
Total size: 339369946 B
Total files: 74
Total blocks (validated): 76 (avg. block size 4465394 B)
*****
UNDER MIN REPL'D BLOCKS: 6 (7.894737 %)
MINIMAL BLOCK REPLICATION: 1
CORRUPT FILES: 6
MISSING BLOCKS: 6
MISSING SIZE: 1267769 B
*****
Minimally replicated blocks: 70 (92.10526 %)
Over-replicated blocks: 0 (0.0 %)
Under-replicated blocks: 70 (92.10526 %)
Mis-replicated blocks: 6 (7.894737 %)
Default replication factor: 3
Average block replication: 1.8421053
Missing blocks: 6
Corrupt blocks: 0
Missing replicas: 98 (38.28125 %)
Blocks queued for replication: 0

Erasure Coded Block Groups:
Total size: 0 B
Total files: 0
Total block groups (validated): 0
Minimally erasure-coded block groups: 0
Over-erasure-coded block groups: 0
Under-erasure-coded block groups: 0
Unsatisfactory placement block groups: 0
Average block group size: 0.0
Missing block groups: 0
Corrupt block groups: 0
Missing internal blocks: 0
Blocks queued for replication: 0
FSCK ended at Sat Oct 30 18:39:21 CEST 2021 in 60 milliseconds

```

Como se puede ver, el número de bloques que aparecen ahora como *under-replicated* es igual a 70. Esto es un número de bloques *under-replicated* muy superior al anterior, debido a los nodos que hemos matado. La salida de este comando también nos muestra que el *filesystem* ahora está corrupto.

Tras esto, probamos a hacer un **get** del fichero **random_words.txt.bz2** para ver si se realiza de forma correcta. Para esto hacemos uso del comando `hdfs dfs -get /user/luser/libros/random_words.txt.bz2` que nos trae este fichero del HDFS a local. La salida de la ejecución de este comando es la siguiente:


```

hdadmin@namenode: ~
Archivo Acciones Editar Vista Ayuda
tcdm@tcdm-docker: ~ hdadmin@namenode: ~
hdadmin@namenode:~$ ls
hadoop-3.3.1
hdadmin@namenode:~$ hdfs dfs -get /user/luser/libros/random_words.txt.bz2 .
hdadmin@namenode:~$ ls
hadoop-3.3.1 random_words.txt.bz2
hdadmin@namenode:~$

```

Como se puede ver el **get** de este fichero sí que se ejecutó de forma correcta. De hecho, si accedemos a la interfaz gráfica del NameNode podemos ver como los ficheros que están corruptos son los siguientes:



Seguimos ahora los pasos indicados en la práctica 1 para añadir un nuevo datanode, el **datanode6**. Una vez hemos añadido este datanode de forma correcta volvemos a ejecutar el comando `hdfs fsck /` para realizar un chequeo de todo el *filesystem*. De esta manera podremos ver como los bloques se van replicando en el nuevo datanode que acabamos de añadir al clúster. Lo primero de todo será comprobar en el datanode6, el cual acabamos de crear, que los demonios están funcionando de forma correcta. Esto lo hacemos mediante el comando `jps`:

```

root@datanode6: /
Archivo Acciones Editar Vista Ayuda
hdadmin@namenode: ~/hadoop/etc/hadoop root@datanode6: /
root@datanode6:/# jps
80 NodeManager
33 DataNode
245 Jps
root@datanode6:/#

```

Después, desde la interfaz web del NameNode podemos ver como los bloques se van replicando en el datanode6:

In operation

DataNode State

All

Show 25 entries

Search:

Node	Http Address	Last contact	Last Block Report	Used	Non DFS Used	Capacity	Blocks	Block pool used	Version
✓/rack1 /datanode5:9866 (172.18.0.4:9866)	http://datanode5:9864	1s	6m	325.43 MB	11.15 GB	29.4 GB	<div><div></div></div> 70	325.43 MB (1.08%)	3.3.1
✓/rack2 /datanode3:9866 (172.18.0.3:9866)	http://datanode3:9864	1s	6m	325.42 MB	11.15 GB	29.4 GB	<div><div></div></div> 70	325.42 MB (1.08%)	3.3.1
✓/rack1 /datanode6:9866 (172.18.0.5:9866)	http://datanode6:9864	1s	6m	324.06 MB	11.15 GB	29.4 GB	<div><div></div></div> 63	324.06 MB (1.08%)	3.3.1

Showing 1 to 3 of 3 entries

Previous

1

Next

Finalmente, el número de bloques en el datanode6 iguala el número de bloques en los datanodes que no fueron detenidos, por lo que se completa el replicado de los bloques:

In operation

DataNode State: All Show: 25 entries Search:

Node	Http Address	Last contact	Last Block Report	Used	Non DFS Used	Capacity	Blocks	Block pool used	Version
✓/rack1 /datanode5:9866 (172.18.0.4:9866)	http://datanode5:9866	0s	7m	325.43 MB	11.15 GB	29.4 GB	<div><div></div></div> 70	325.43 MB (1.08%)	3.3.1
✓/rack2 /datanode3:9866 (172.18.0.3:9866)	http://datanode3:9866	0s	7m	325.42 MB	11.15 GB	29.4 GB	<div><div></div></div> 70	325.42 MB (1.08%)	3.3.1
✓/rack1 /datanode6:9866 (172.18.0.5:9866)	http://datanode6:9866	0s	7m	324.99 MB	11.15 GB	29.4 GB	<div><div></div></div> 70	324.99 MB (1.08%)	3.3.1

Showing 1 to 3 of 3 entries

Previous 1 Next

También lo podemos comprobar haciendo un chequeo del *filesystem* mediante el comando `hdfs fsck /`:

```

hadoop@namenode: ~
Archivo Acciones Editar Vista Ayuda
tcdm@tcdm-docker: ~
hadoop@namenode: ~

Total size: 339369946 B
Total files: 74
Total blocks (validated): 76 (avg. block size 4465394 B)
*****
UNDER MIN REPL'D BLOCKS: 6 (7.894737 %)
MINIMAL BLOCK REPLICATION: 1
CORRUPT FILES: 6
MISSING BLOCKS: 6
MISSING SIZE: 1267769 B
*****
Minimally replicated blocks: 70 (92.10526 %)
Over-replicated blocks: 0 (0.0 %)
Under-replicated blocks: 4 (5.263158 %)
Mis-replicated blocks: 6 (7.894737 %)
Default replication factor: 3
Average block replication: 2.7631578
Missing blocks: 6
Corrupt blocks: 0
Missing replicas: 28 (10.9375 %)
Blocks queued for replication: 0

Erasure Coded Block Groups:
Total size: 0 B
Total files: 0
Total block groups (validated): 0
Minimally erasure-coded block groups: 0
Over-erasure-coded block groups: 0
Under-erasure-coded block groups: 0
Unsatisfactory placement block groups: 0
Average block group size: 0.0
Missing block groups: 0
Corrupt block groups: 0
Missing internal blocks: 0
Blocks queued for replication: 0
FSCK ended at Sat Oct 30 19:31:56 CEST 2021 in 78 milliseconds

```

Como se puede ver en esta captura, el número de bloques *under-replicated* vuelve a ser el que era antes de matar a los datanodes.

3 PROBAR EL USO DE CÓDIGOS DE BORRADO (ERASURE CODES)

Vemos las políticas de EC disponibles mediante el uso del comando `hdfs ec -listPolicies`:

```

hdadmin@namenode: ~
Archivo Acciones Editar Vista Ayuda
tcdm@tcdm-docker: ~ hdadmin@namenode: ~
hdadmin@namenode:~$ hdfs ec -listPolicies
Erasure Coding Policies:
ErasureCodingPolicy=[Name=RS-10-4-1024k, Schema=[ECSchema=[Codec=rs, numDataUnits=10, numParityUnits=4]], CellSize=1048576, Id=5], State=DISABLED
ErasureCodingPolicy=[Name=RS-3-2-1024k, Schema=[ECSchema=[Codec=rs, numDataUnits=3, numParityUnits=2]], CellSize=1048576, Id=2], State=DISABLED
ErasureCodingPolicy=[Name=RS-6-3-1024k, Schema=[ECSchema=[Codec=rs, numDataUnits=6, numParityUnits=3]], CellSize=1048576, Id=1], State=ENABLED
ErasureCodingPolicy=[Name=RS-LEGACY-6-3-1024k, Schema=[ECSchema=[Codec=rs-legacy, numDataUnits=6, numParityUnits=3]], CellSize=1048576, Id=3], State=DISABLED
ErasureCodingPolicy=[Name=XOR-2-1-1024k, Schema=[ECSchema=[Codec=xor, numDataUnits=2, numParityUnits=1]], CellSize=1048576, Id=4], State=DISABLED
hdadmin@namenode:~$
  
```

Como se puede ver, tenemos activada la política de EC por defecto, de nombre 'RS-6-3-1024k'. Tras haber iniciado los dockers necesarios para tener al menos 5 datanodes (en realidad iniciamos los 6 datanodes que tenemos creados para hacer uso del clúster al completo), en el NameNode, como usuario hdadmin habilitamos la política 'RS-3-2-1024k' ejecutando el comando `hdfs ec -enablePolicy -policy RS-3-2-1024k`.

Después de ejecutar este comando, creamos una carpeta **/user/grandes** en el HDFS mediante el comando `hdfs dfs -mkdir /user/grandes` para después indicar que se le aplique la política de EC 'RS-3-2-1024k' mediante el comando `hdfs ec -setPolicy -path /user/grandes -policy RS-3-2-1024k`.

Las políticas EC (*Erasure Codes*), como dice en la documentación de hadoop, se nombran siguiendo el siguiente esquema: **codec-num data blocks-num parity blocks-cell size**. Estas políticas encapsulan como codificar y decodificar los ficheros.

Ahora para ver que funciona seguiremos los pasos indicados en el enunciado. En primer lugar, con el comando `hdfs dfsadmin -report` comprobamos el espacio ocupado en DFS:

```

hdadmin@namenode: ~
Archivo Acciones Editar Vista Ayuda
tcdm@tcdm-docker: ~ hdadmin@namenode: ~
hdadmin@namenode:~$ hdfs dfsadmin -report
Configured Capacity: 189416521728 (176.41 GB)
Present Capacity: 110072156160 (102.51 GB)
DFS Remaining: 109042606080 (101.55 GB)
DFS Used: 1029550080 (981.86 MB)
DFS Used%: 0.94%
Replicated Blocks:
  Under replicated blocks: 4
  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: 172.18.0.3:9866 (datanode1.hadoop-cluster)
Hostname: datanode1
Rack: /rack2
Decommission Status : Normal
Configured Capacity: 31569420288 (29.40 GB)
DFS Used: 67923968 (64.78 MB)
Non DFS Used: 11700494336 (10.90 GB)
DFS Remaining: 18173767680 (16.93 GB)
DFS Used%: 0.22%
DFS Remaining%: 57.57%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)

```

Como se puede ver, el espacio ocupado en DFS es de 981,86MB, siendo esta memoria el 0,94% del espacio total del DFS.

Ejecutamos `hdfs dfs -get /user/luser/libros/random_words.txt.bz2` para obtener ese fichero grande desde HDFS. Después borramos ese fichero de HDFS mediante el comando `hdfs dfs -rm /user/luser/libros/random_words.txt.bz2` y borramos la papelera mediante el comando `hdfs dfs -expunge`. Ponemos el fichero en la carpeta **/user/grandes** mediante el comando `hdfs dfs -put random_words.txt.bz2 /user/grandes` y comprobamos de nuevo el espacio ocupado en el DFS:

```

hdadmin@namenode: ~
Archivo Acciones Editar Vista Ayuda
tcdm@tcdm-docker: ~ hdadmin@namenode: ~
hdadmin@namenode:~$ hdfs dfsadmin -report
Configured Capacity: 189416521728 (176.41 GB)
Present Capacity: 112388722653 (104.67 GB)
DFS Remaining: 111803015168 (104.12 GB)
DFS Used: 585707485 (558.57 MB)
DFS Used%: 0.52%
Replicated Blocks:
    Under replicated blocks: 4
    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: 172.18.0.3:9866 (datanode1.hadoop-cluster)
Hostname: datanode1
Rack: /rack2
Decommission Status : Normal
Configured Capacity: 31569420288 (29.40 GB)
DFS Used: 116535303 (111.14 MB)
Non DFS Used: 11191820281 (10.42 GB)
DFS Remaining: 18633830400 (17.35 GB)
DFS Used%: 0.37%
DFS Remaining%: 59.02%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)

```

Ahora se puede ver como el espacio total usado del DFS es de 558,57MB, que es menor que el espacio usado anteriormente (981,86MB). Esto se debe a que el fichero **random_words.txt.bz2**, que ocupa 315MB aproximadamente, antes se dividía en 5 bloques, cada uno de los cuales se replicaba 3 veces, por lo que ocupaba un total de 15 bloques. Mediante el uso de esta política de EC el fichero se divide en 3 celdas de datos más 2 celdas de paridad, además de ir codificado mediante el codificado Reed-Salomon, por lo que ocupa menos espacio en el DFS.