

Pendente: domingo, 29 de outubro de 2023, 23:59

Gestión de HDFS

En esta práctica veremos algunos comandos para gestionar el HDFS.

Primera parte: Ver como se dividen y replican los ficheros

1. En el nameNode, como usuario luser, crea un fichero grande con el siguiente comando:

```
dd if=/dev/urandom of=fichero_grande bs=1M count=350
```

2. Mueve ese fichero al HDFS (con `hdfs dfs -moveFromLocal`)
3. Accede al interfaz web del NameNode y busca el fichero. Responde a las siguientes preguntas:
 - ¿En cuántos bloques se ha dividido el fichero?
 - Para cada uno de estos bloques ¿en que DataNodes se encuentran sus réplicas?
4. Obtén la misma información usando el comando `hdfs fsck` con las opciones adecuadas (busca en la ayuda de `hdfs fsck` cuáles son esas opciones).

Segunda parte: Probar el comando **hdfs dfsadmin**

1. En el NameNode, como usuario `hdadmin`, crea un directorio en HDFS y ponle una cuota de solo 4 ficheros. Comprueba cuántos ficheros puedes copiar a ese directorio. Explica a qué se debe este comportamiento.

Tercera parte: Probar el comando **hdfs fsck**

1. En el NameNode (como usuario `hdadmin`) haz un chequeo de todo el HDFS, y comprueba si te da errores:
 - Intenta determinar las causas de los posibles errores y resuélvelos
2. Detén datanodes de forma brusca, parando los dockers, (sin detener los demonios, simplemente haciendo p.e. `docker container stop datanode1 datanode3 datanode5`) hasta que te queden **solo 2 datanodes vivos en dos racks diferentes**. Espera unos 10 minutos¹ y comprueba que el comando `hdfs dfsadmin -report` muestra que, efectivamente, solo quedan 2 datanodes activos. ¿Cuántos bloques tiene cada Datanode?
3. Realiza de nuevo el chequeo de disco en el NameNode y comprueba la salida. ¿Cuántos bloques aparecen *under-replicated*? ¿aparecen bloques perdidos y ficheros corruptos? En el caso de que haya bloques perdidos, determina a qué fichero/s corresponden (busca en la ayuda de `hdfs fsck --help` cómo encontrarlos).
4. Usa `hdfs fsck` para chequear la disponibilidad de los bloques del fichero `fichero_grande`. Obtén el número de réplicas de cada bloque y su localización. ¿Es posible recuperar el fichero?
5. Sigue los pasos que vistes en la práctica 1 para añadir un datanode nuevo (`datanode7`). Comprueba, haciendo de nuevo el chequeo, que los datos se replican en el nuevo nodo hasta que no quedan bloques *under-replicated*. ¿Con cuántos bloques se queda el `datanode7`? ¿cuál es el factor de replicación medio?

?

Tercera parte: Probar el uso de **códigos de borrado (erasure codes o EC)**

Para poder utilizar EC en vez de replicación es necesario tener activos como mínimo 5 datanodes. EC se aplica a ficheros nuevos que se guarden en carpetas en las que se haya especificado una política de EC.

Existen diferentes políticas de EC, que garantizan una mayor o menor protección frente a fallos, con menor o mayor ahorro de espacio. Se pueden ver las políticas disponibles con el comando `hdfs ec -listPolicies`

La política que se usa por defecto se define a través de la propiedad `'dfs.namenode.ec.system.default.policy'`, y es `"RS-6-3-1024k"` por defecto.

Lo primero que haremos será crear una carpeta para la que especificaremos una política de EC. Sigue los siguientes pasos:

1. Inicia los dockers que paraste en el apartado anterior (`docker start datanode1 datanode3 datanode5`) y espera hasta que se recuperen todos los bloques del HDFS y este deje de estar corrupto.
2. En el NameNode, como usuario `hdadmin`, comprueba las políticas disponibles con el comando antes indicado
3. Habilita la política `"RS-3-2-1024k"` ejecutando:

```
hdfs ec -enablePolicy -policy RS-3-2-1024k
```

4. Crea una carpeta `/user/grandes` en HDFS, para la que vamos a indicar que se aplique una política EC

5. Aplica la política EC con:

◦ `hdfs ec -setPolicy -path /user/grandes -policy RS-3-2-1024k`

Con esos pasos, aplicamos la política EC indicada en el directorio /user/grandes. Vamos a ver que funciona. Ejecuta los siguientes pasos en el NameNode como hdadmin:

1. Comprueba con `hdfs dfsadmin -report` el espacio ocupado en DFS, y apunta el valor (captura de pantalla). Anota también el número de bloques que tiene cada datanode.
2. Mueve el fichero grande de su localización a la carpeta /user/grandes (da un warning, pero la copia se realiza igual)
3. Comprueba de nuevo el espacio ocupado en DFS y comparalo con el valor que había antes. Compara también el número de bloques en cada Datanode con lo que había antes ¿cuál es la diferencia entre los bloques actuales y los anteriores?

Entrega

1. Un documento que muestre que se han ejecutado los diferentes apartados y las respuestas a las preguntas que se van haciendo. Mostrar capturas de pantalla, incluyendo una explicación y justificación de lo que aparece en las mismas.

[1] El tiempo depende de los parámetros `dfs.namenode.stale.datanode.interval` y `dfs.namenode.heartbeat.recheck-interval`. El primero indica el tiempo sin detectar actividad del DataNode para que el NameNode lo considere en estado stale (por defecto, 30 segundos). Un nodo en estado stale tiene menor prioridad en lecturas y escrituras. El segundo parámetro indica el intervalo de chequeo en busca de DataNodes expirados (valor por defecto, 5 minutos). Un DataNode se pasa al estado Dead cuando el tiempo sin detectar actividad es superior a `dfs.namenode.stale.datanode.interval + 2 * dfs.namenode.heartbeat.recheck-interval`.

Engadir entrega

Estado da entrega

Estado da entrega	Aínda non se realizaron entregas
Estado das cualificacións	Sen cualificar
Tempo restante	20 días 13 horas restante
Última modificación	-
Comentarios a entrega	<div>► Comentarios (0)</div>