



36018173. CSIFC90. MP5075. Big Data Aplicado. 2023-2024. (Grupo A)

Parcial MARZO24

Nombre completo y DNI

Leer antes de empezar.

- La duración del examen es de 2 horas con 5 minutos de cortesía.
- Al acabar el examen debes firmar en la hoja de entrega.
- Ni se permitirá la firma en la hoja de entrega ni se recogerán exámenes fuera de tiempo.
- Durante esta prueba escrita puedes utilizar libremente todo tu material escrito y digital.
- No estará permitido comunicarse con el resto de alumnado por ningún medio.
- La prueba escrita debe permanecer inmutable por lo que es obligatorio usar bolígrafo permanente y no escribir en lápiz, usar corrector o cualquier otro método que pueda poner en duda una alteración posterior.
- Es necesario usar lenguaje técnico.
- Se puede solicitar hojas extra si se necesitan.
- El espacio disponible es suficiente para contestar cada pregunta.
- Las preguntas tipo test tienen una única respuesta válida. Preguntas incorrectas puntúan en negativo.





1.- (0,25) Desde el punto de vista del almacenamiento en Big Data:

- ☐ Está basado en RAID5
- ☐ **Se espera que el hardware falle**
- ☐ Está constituido por hardware potente y muy especializado
- ☐ Se tiene en cuenta defensas ante virus informáticos

2.- (0,25) A nivel de sistema operativo, para cambiar de nombre a un nodo:

- ☐ Solo se puede hacer durante la instalación del sistema operativo.
- ☐ **Editamos el archivo /etc/hostname y reiniciamos**
- ☐ Podemos usar el comando adduser con los parámetros adecuados.
- ☐ Editamos el archivo /var/lib/hosts y reiniciamos

3.- (0,25) Solo el usuario “hdfsuser” de un determinado nodo va a tener permisos y privilegios de sistema operativo en Hadoop. Debemos escribir las variables de entorno en:

- ☐ /etc/bash
- ☐ /home/hdfsuser/environment
- ☐ **/home/hdfsuser/.bashrc**
- ☐ /etc/environment

4.- (0,25) Con el comando “ssh-copy-id” podemos

- ☐ Copiar un fichero de manera cifrada desde local al sistema de fichero HDFS en remoto.
- ☐ Copiar e instalar una llave privada en un nodo remoto.
- ☐ **Copiar e instalar una llave pública en un nodo remoto.**
- ☐ Generar una nueva pareja de claves asimétricas

5.- (0,25) Al usar claves asimétricas con un nombre de archivo personalizado debemos

- ☐ No hay que hacer nada, ssh conectará sin pedir contraseña
- ☐ Hay que añadir la ruta al archivo /etc/ssh/sshd_config
- ☐ Hay que añadir el parámetro “-i” con la ruta en la llamada al comando “ssh”
- ☐ **Hay que añadir la ruta al archivo \$HOME/.ssh/config (hadoop no quiere problemas)**

6.- (0,25) Para consultar con certeza cual es la puerta de enlace de un nodo ejecutarías

- ☐ cat /etc/hosts
- ☐ cat /etc/netplan/01-network-manager-all.yaml
- ☐ **ip route**
- ☐ Ip address



7.- (0,25) La propiedad de HDFS en la que indicamos el valor de replicación deseado se debe guardar, como mínimo, en:

- ☐ core-site.xml en namenode
- ☐ core-site.xml en datanodes
- ☐ **hdfs-site.xml en namenode**
- ☐ hdfs-site.xml en datanodes

8.- (0,25) Para que funcione el clúster HDFS, todos los archivos hdfs-site.xml de los datanodes deben, obligatoriamente tener:

- ☐ La propiedad dfs.datanode.data.dir
- ☐ La propiedad dfs.namenode.name.dir
- ☐ La propiedad dfs.replication
- ☐ **No hace falta ninguna propiedad, tomarán valores por defecto.**

9.- (0,25) Para listar el contenido de la carpeta “/javi” dentro del sistema de ficheros HDFS usaremos el comando:

- ☐ ssh hdfs | ls -la /javi
- ☐ cat /mnt/discogrande/javi
- ☐ **hdfs dfs -ls /javi**
- ☐ ls -la /mnt/discogrande/javi

10.- (0,25) El contenido del archivo “workers”

- ☐ Solo se usa en el namenode y contiene todos los nodos del clúster
- ☐ Se usa en todos los datanodes y solo puede contener localhost y el nombre del nodo
- ☐ Se usa en todos los nodos y contiene todos los datanodes
- ☐ **Está en todos los nodos pero no es obligatorio usarlo**

11.- (0,25) Los archivos subidos a HDFS:

- ☐ Solo pueden ocupar 128 megabytes
- ☐ **Solo se pueden editar para anexar contenido, no para modificar.**
- ☐ No tienen permisos de escritura, lectura o ejecución.
- ☐ Es mejor muchos archivos de pequeño tamaño que uno grande.

12.- (0,75) Al levantar un clúster HDFS usando el script “start-dfs.sh” desde el namenode obtenemos la siguiente salida. Indica los pasos y comandos que usarías para diagnosticar y ofrecer una posible solución.

```
hadoop@equipo22:~$ start-dfs.sh
Starting namenodes on [equipo22]
Starting datanodes
equipo17: hadoop@equipo17: Permission denied (publickey,password).
Starting secondary namenodes [equipo22]
hadoop@equipo22:~$
```

✓/default-rack/equipo12:9866 (10.0.0.12:9866)	http://equipo12:9864	1s	4m	129.04 MB	76.04 MB	897.68 GB	<div></div>	1	129.04 MB (0.01%)	3.3.6
✓/default-rack/equipo14:9866 (10.0.0.14:9866)	http://equipo14:9864	2s	4m	185.77 MB	76.04 MB	897.68 GB	<div></div>	2	185.77 MB (0.02%)	3.3.6
✓/default-rack/equipo09:9866 (10.0.0.9:9866)	http://equipo09:9864	1s	4m	32 KB	44 KB	897.61 GB	<div></div>	0	32 KB (0%)	3.3.6
✗/default-rack/equipo17:9866 (10.0.0.17:9866)			Wed Mar 13 20:24:14 +0100 2024							

Showing 1 to 18 of 18 entries

Previous 1 Next

Problema de llaves asimétricas (publickey).

Intentaría conectar desde el namenode al equipo17 y viceversa para saber si el problema es un una sola dirección o ambas.

Comprobaría si las llaves públicas de ambos están copiadas en el otro, volviéndolas a instalar. Aquí es donde está el problema.

Probaría a levantar el datanode a mano para ver si se incorpora. O añadiría el equipo17 al workers y actualizaría o apagar/encender clúster.

13.- (0,75) Un técnico nos indica que acaba de configurar el “equipo17” pero que no consigue levantarlo como datanode.

```
hadoop@equipo17:~$ hdfs --daemon start datanode
ERROR: Cannot set priority of datanode process 2560775
hadoop@equipo17:~$
```

```
GNU nano 4.8      ./hadoop/etc/hadoop/core-site.xml      Modificado
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://equipo22:9000</value>
  </property>
</configuration>
```

```
GNU nano 4.8      ./hadoop/etc/hadoop/hdfs-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/home/hadoop/discogrande/datanode</value>
  </property>
</configuration>
```

Hay un error en el cierre de la etiqueta value del core-site.xml

14.- (0,6) El equipo14 está ya “decomisionado” pero aún tiene latidos. En este nodo se guardan dos bloques. ¿Qué pasaría si apagamos ahora mismo este nodo “equipo14”?

✓/default-rack/equipo07:9866 (10.0.0.7:9866)	http://equipo07:9866	1s	5m	129.04 MB	76.04 MB	897.68 GB	<div></div>	1	129.04 MB (0.01%)	3.3.6
✓/default-rack/equipo15:9866 (10.0.0.15:9866)	http://equipo15:9866	2s	5m	32 KB	76.04 MB	897.68 GB	<div></div>	0	32 KB (0%)	3.3.6
✓/default-rack/equipo12:9866 (10.0.0.12:9866)	http://equipo12:9866	1s	5m	129.04 MB	76.04 MB	897.68 GB	<div></div>	1	129.04 MB (0.01%)	3.3.6
✓/default-rack/equipo09:9866 (10.0.0.9:9866)	http://equipo09:9866	1s	5m	32 KB	44 KB	897.61 GB	<div></div>	0	32 KB (0%)	3.3.6
⚠/default-rack/equipo14:9866 (10.0.0.14:9866)	http://equipo14:9866	2s	5m	185.77 MB	76.04 MB	897.68 GB	<div></div>	2	185.77 MB (0.02%)	3.3.6

Solamente pasaría a decomisionado y muerto. No desencadena ninguna copia de bloques porque durante el proceso de decomisionado ya sean replicado los bloques durante la preparación del decomiso.

15.- (0,75) Indica el comando CLI (por terminal) con el consigues la siguiente salida

```
Configured Capacity: 14461596139520 (13.15 TB)
Present Capacity: 13725287907328 (12.48 TB)
DFS Remaining: 13723815632896 (12.48 TB)
DFS Used: 1472274432 (1.37 GB)
DFS Used%: 0.01%
Replicated Blocks:
    Under replicated blocks: 0
    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 (18):

Name: 10.0.0.10:9866 (equipo10)
Hostname: equipo10
Decommission Status : Normal
Configured Capacity: 963801133056 (897.61 GB)
DFS Used: 59523072 (56.77 MB)
Non DFS Used: 45056 (44 KB)
DFS Remaining: 914707632128 (851.89 GB)
DFS Used%: 0.01%
DFS Remaining%: 94.91%
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: Wed Mar 13 21:43:50 CET 2024
Last Block Report: Wed Mar 13 21:32:29 CET 2024
Num of Blocks: 1

Name: 10.0.0.11:9866 (equipo11)
Hostname: equipo11
Decommission Status : Normal
Configured Capacity: 963801133056 (897.61 GB)
```

Hdfs dfsadmin -report

16.- (0,75) Imagina una red mundial colaborativa formada por miles de particulares en la que se monta un clúster HDFS. Los nodos son los PCs de los propios particulares en los que simplemente hay un segundo disco muy pequeño (podría ser un USB de 1 gigabyte como mucho) dedicado a almacenamiento HDFS. Como sabes, en HDFS se envían latidos regularmente. El tiempo entre latidos y de reintentos antes de declarar un nodo como muerto se puede configurar.

Justifica detalladamente con tus palabras cuantos segundos deberían pasar hasta declarar un nodo muerto.

El problema de las caídas de los nodos es la gran sobrecarga que conlleva buscar y replicar los datos que estaban allí. En este caso los discos son tan pequeños, tan solo caben 8 bloques, que el sobre coste de esas réplicas no es muy grande.

La detección de un nodo caído se hace rápidamente, basta con que no conteste a un ping pero para declararlo muerto podríamos dar algo más de margen. En la práctica comprobamos que 10 segundos parece un tiempo razonable para descartar un corte puntual y asumir que ese equipo se ha apagado hasta mañana.

Yo pondría 10 segundos.

17.- (0,6) Explica con tus palabras y con algún ejemplo, para que se usen las propiedades “dfs.hosts” y “dfs.hosts.exclude”.

El primero indica la ruta donde está el archivo que contiene las IPs o nombres de los datanode. Típicamente usaremos el archivo workers pero no necesariamente debe ser este, podemos usar cualquiera.

El segundo indica la ruta donde está el archivo que contiene las IPs o nombres de los datanodes que queremos decomisar. Mientras una IP o un nombre figure en este archivo, el namenode se encargará de que todos los bloques de ese nodo se repliquen. Momentáneamente tendremos más réplicas que las indicadas por el administrador. Además tampoco enviará a ese nodo nuevos bloques ni peticiones de lectura.

Como ejemplo, al crear el clúster podemos añadir todos los datanodes en el archivo indicado por la propiedad `dfs.hosts`. Automáticamente aparecerán en el informe web o por CLI. Si tienen latido los veremos en verde y si no responden los veremos en rojo.

Si en algún momento queremos apagar un nodo para tareas de mantenimiento pondremos su nombre o esa IP en el archivo indicado por la ruta del `dfs.hosts.exclude` y refrescaremos los nodos con un `hdfs dfsadmin -refreshNodes` (o todo el clúster)

Durante unos segundos el namenode se encargará de replicar los datos de este nodo. Cuando acabe dejará de mandarle peticiones de ningún tipo y lo pasará a estado “Decomisado”. Entonces lo podremos apagar sin consecuencias.

Al acabar el mantenimiento. Editamos el archivo indicado por `dfs.hosts.exclude` y quitamos este nodo. Refrescamos y encendemos el nodo lanzando el datanode explícitamente (o desde namenode)

Si alguna vez queremos dar de baja definitivamente un datanode y quitarlo de la lista será suficiente con eliminarlo del archivo indicado por `dfs.hosts`

18.- (0,6) Explica con tus palabras qué secuencia de acontecimientos se suceden cuando se sube un archivo de 200 megabytes a un clúster HDFS.

El namenode recibe la petición del cliente y le pide partirlo en bloques de 128 megas. Este archivo genera dos bloques, uno de 128 megas y el otro de 72 megas.

El siguiente paso consiste en que el namenode le indica al cliente en qué datanode debe copiar el bloque. Será el cliente el que lo copie directamente en el datanode indicado por el namenode.

En paralelo, el namenode ordenará al datanode que copie el bloque en tantos otros nodos como para satisfacer la propiedad de replication, normalmente 3.

Con el siguiente bloque, el de 72, se repite el proceso. Namenode indica al cliente en qué datanode copiar el bloque.

En paralelo, el namenode ordenará la copia del bloque en otros nodos para satisfacer el nivel de réplica.

Cuando la operación acabe se dará la confirmación de escritura al cliente.

19.- (0,6) ¿Como añadirías un nuevo datanode a un clúster HDFS que ya existe?

Configuraría todo lo relacionado con el sistema operativo como IP, llaves asimétricas y su intercambio con todos los nodos (y localhost), usuarios, permisos, rutas de almacenamiento.

Descargaría y descomprimiría hadoop, confirmaría permisos y variables de entorno.

Editaría core-site.xml con los datos del namenode. Editaría hdfs-site con los datos relacionados con la ruta de almacenamiento persistente.

Me aseguraría de incluir en el archivo indicado por el dfs.hosts del namenode, la ip o nombre de este nuevo nodo.

Arrancaría a mano el datanode con un `hdfs -daemon start datanode`. Si aún no se viera haría un `refreshNodes` para confirmar si hay algún problema.

20.- (1,25) Explica detalladamente con tus propias palabras los motivos técnicos por los que los “jumbo chunks” serían un problema de eficiencia en un MongoDB que usa un clúster HDFS como soporte de almacenamiento de datos. Justifica si es una razón suficiente para descartar la combinación (MongoDB+HDFS) o tienen otras ventajas que lo hacen interesante en otros escenarios.

Los chunks son los trozos en los que se parte una colección de mongodb en sharding. Cada uno de estos chunks puede almacenar 128 megas y llegado ese momento se parte en otros dos usando unos nuevos límites de los valores de clave de fragmentación. Hay ocasiones en las que la clave de fragmentación tiene poca cardinalidad y no es posible generar nuevos chunks con límites distintos así que la única solución pasa porque el chunk crezca por encima de los 128 megas. Estos son los llamados jumbo chunks.

En principio usar HDFS con Sharding de MongoDB parece muy buena idea porque el tamaño del bloque HDFS y el chunk del sharding es de 128 ambos. El problema con los jumbo chunks es que se convertirían en 2 o más bloques hdfs que podrían estar en nodos distintos haciendo que la consulta fuera extremadamente lenta.

*** Otra opción es preferir seguridad y replicación a costa de velocidad.



21.- (0,6) De manera breve explica cómo crear un clúster HDFS. Puedes suponer que los nodos ya están correctamente configurados hasta el punto de poder comunicarse entre ellos mediante claves asimétricas.

- Decidiría qué nodo será el namenode.
- Editaría core-site.xml con la ruta al namenode (extra nombre propiedad?)
- Copiaría o modificaría este archivo en todos los nodos.
- Editaría hdfs-site en el namenode con las propiedades de almacenamiento persistente de la tabla de bloques y otros aspectos opcionales como dfs.hosts, dfs.hosts.exclude, replication, Interval, heartbeat,... (extra propiedades?)
- Editaría hdfs-site en todos los datanodes modificando la ruta del almacenamiento persistente.
- Tendría que dar formato al namenode (extra comando?)
- OPCIONAL añadir los datanodes al archivo indicado por dfs.hosts (y si no es el mismo también en workers)
- Arrancaría el namenode y después todos los datanodes
- Confirmaría por comando o interfaz web que todo está ok.

