

Práctica 1 - Hadoop

Francisco Javier Sáez Maldonado, José Antonio Álvarez Ocete

Cuestiones planteadas

Pregunta 1.1. ¿ Qué ficheros ha modificado para activar la configuración del HDFS? ¿ Qué líneas ha sido necesario modificar?

Hemos modificado el fichero `/opt/hadoop-2.8.1/etc/hadoop/hadoop-env.sh` añadiendo la línea `export JAVA_HOME= /usr/lib/jvm/jre-1.7.0-openjdk` para especificar la instalación de Java que queremos utilizar

Como se explica en <https://stackoverflow.com/questions/17569423/what-is-best-way-to-start-and-stop-hadoop-ecosystem-with-command-line>, el script `stop-all.sh` detiene todos los daemons de Hadoop a la vez, pero está obsoleto. En lugar de eso es recomendable parar los daemons de HDFS y YARN por separado en todas las máquinas utilizando `stop-dfs.sh` y `stop-yarn.sh`.

A continuación, para instalar Hadoop pseudo-distribuido, hemos modificado el fichero `/opt/hadoop/etc/hadoop/core-site.xml`:

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

Y añadimos al fichero `/opt/hadoop/etc/hadoop/hdfs-site.xml` lo siguiente:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

A continuación, hemos realizado la instalación del sistema pseudo-distribuido usando YARN, así que hemos modificado los siguientes ficheros **para configurar el uso de YARN (no de HDFS)**. `etc/hadoop/mapred-site.xml`:

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

Y también `etc/hadoop/yarn-site.xml`:

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>

  <property>
    <name>yarn.nodemanager.aux-services.mapreduce_shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
</configuration>
```

Ejercicio 1.2: Para pasar a la ejecución de Hadoop sin HDFS, ¿ es suficiente con parar el servicio con `stop-dfs.sh`? ¿ Cómo se consigue ?

Proceso

Describiremos a continuación los pasos que se han ido siguiendo para obtener los resultados finales de la práctica.

Instalamos java

```
yum install java-1.8.0-openjdk-devel
```

Cambiamos la versión por defecto que se usaba en las transparencias

Incluso si instalamos la versión 1.7 se instalará la versión 1.8. Hemos de cambiar los paths a las versión 1.8 para que funcione correctamente:

```
export JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk
```

Editamos el archivo `/opt/hadoop/etc/hadoop/hadoop.env.sh`:

```
export JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk
```

Compilación

Tomamos el código que se ha proporcionado y lo

```
bin/hadoop fs -cat /user/bigdata/compilar.bash | exec bash -s WordCount
```

Para ejecutar

Primero hay que iniciar el NameNode y el DataNode:

```
sbin/start-dfs.sh
```

Iniciar el ResourceManager y el NodeManager:

```
sbin/start-yarn.sh
```

Recuerda que hemos de subir el archivo utilizando:

```
/opt/hadoop/bin/hdfs dfs -put Quijote.txt /user/root
```

Lanzamos nuestro trabajo de MapReduce:

```
sudo /opt/hadoop/bin/hadoop jar WordCount.jar uam.WordCount Quijote.txt output/
```

Obtenemos en el directorio output la salida. En concreto, dos archivos:

- Un archivo SUCCESS indicando que la tarea ha sido exitosa.
- Un archivo part-r-00000 que tiene la salida del programa que queríamos ejecutar.

Mostramos una parte del fichero para mostrar parte de la salida (la salida completa se puede encontrar en este archivo).

```
"Tablante", 1
"dichosa    1
"el 8
"y 1
";Oh, 1
(Y 1
(a 1
(al 1
(como 1
(creyendo 1
(de 2
(habiéndose 1
(por 2
(porque 2
(pues 1
(que 21
```

Como podemos comprobar, las palabras quedan con ciertos símbolos de puntuación que no nos interesa que estén para realizar el conteo correcto de palabras. Para arreglar esto, solo debemos cambiar una línea en la función Map del archivo WordCount.java. En concreto, la dejamos de la siguiente forma:

```
StringTokenizer itr = new StringTokenizer(value.toString().
    toLowerCase().replaceAll("[^a-z ]", ""));
```

como vemos, hemos pasado las palabras a minúsculas usando toLowerCase y luego hemos eliminado todo aquello que no sean letras usando replaceAll.