

Práctica 1. Hadoop y Spark

Blanca Cano Camarero y Iker Villegas Labairu

Madrid 8 de octubre del 2022

Índice

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Práctica 1 Hadoop y Spark | 3 |
| Instalación | 3 |
| Preguntas expuestas en las diapositivas | 4 |
| Ejercicio 1.1 | 4 |
| ¿Qué ficheros ha modificado para activar la configuración del HDFS? | 4 |
| ¿Qué líneas ha sido necesario modificar? | 4 |
| Ejercicio 1.2 | 4 |
| Ejercicio 3.1 | 5 |
| Modificar el ejemplo de WordCount que hemos tomado como partida, para que no tenga en cuenta signos de puntuación ni las mayúsculas ni las minúsculas volver a ejecutar la aplicación | 5 |

Práctica 1 Hadoop y Spark

Instalación

Para la instalación se ha seguido estrictamente el tutorial de clase, además para agilizar el proceso y puesto que en ambiente actual es interesante el uso de contenedores, hemos realizado un script de bash, para automatizar el proceso de instalación hasta la parte de ssh en un contenedor de docker con CentOS:7:

```
1 echo "Task 1: Updating system"
2 yum -y update && yum clean all
3 echo "Task 2: Installing packages"
4 yum -y install wget
5 yum -y install which
6 yum -y install rsync
7 yum -y install java-1.7.0-openjdk
8 echo "Task 3: Hadoop installation (may take a while)"
9 wget https://archive.apache.org/dist/hadoop/common/hadoop-2.8.1/hadoop
  -2.8.1.tar.gz
10 tar xvzf hadoop-2.8.1.tar.gz
11 mv hadoop-2.8.1 opt/
12 cd opt/ && ln -s hadoop-2.8.1 hadoop
13 export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk
14 echo 'export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk' >> ./hadoop/etc/
  hadoop/hadoop-env.sh
15 # Comando de prueba
16 #./opt/hadoop/bin/hadoop
17
18 yum -y install openssh-server openssh-clients
```

El respectivo Dockerfile sería el siguiente:

```
1 FROM centos:7
2 COPY ./initial.sh /
3 COPY ./material /
4 RUN bash ./initial.sh
5 EXPOSE 50070
```

Donde la carpeta material contiene:

```
1 (main)> tree material/
2 material/
3 |--WordCount_final.java
4 |-- quijote.txt
```

Preguntas expuestas en las diapositivas

Las diapositivas con la guía de instalación) estaba salpicada de preguntas, vamos a proceder a reponerlas:

preguntas sobre la instalación de Hadoop Pseudo distribuida

1. **¿En qué directorios del HDFS se copian los ficheros** (Diapositiva 41)
Se copian en el directorio `/user/bigdata`.
2. **¿Qué ocurre si no hubieramos creado el directorio?** (Diapositiva 41) Obtendríamos un error del estilo `No such file or directory`.

Ejercicio 1.1

¿Qué ficheros ha modificado para activar la configuración del HDFS?

Bajo supuesto de que la instalación anterior es correcta y las variables de entorno (la versión de Java) ya se hayan exportado.

Los ficheros que debemos de instalar son

`/hadoop/hdfs/hdfs-site.xml`

Al cual se le ha añadido la configuración (copiar de la pag 38)

¿Qué líneas ha sido necesario modificar?

Ha sido necesario añadir el valor de la variable tal escrita como.

Es necesario además ejecutar el comando `bin/dfs namenode -format` que afecta a ciertos ficheros así que estos también se ven modificados.

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? Sí, hay que ejecutar `stop-dfs.sh` esto es con el comando `sbin/stop-dfs.sh` ya que para todos los servicios, en particular la ejecución de Hadoop con HDFS.

Ejercicio 3.1

Modificar el ejemplo de WordCount que hemos tomado como partida, para que no tenga en cuenta signos de puntuación ni las mayúsculas ni las minúsculas volver a ejecutar la aplicación

Para conseguir esto hemos modificado el método map de la siguiente manera

```
1 public void map(Object key, Text value, Context context) throws
   IOException, InterruptedException {
2     String minimized_value = value.toString().toLowerCase();
3     String processed_value = minimized_value.replaceAll("[^áéíóúüñá-
        z0-9]+", "");
4     StringTokenizer itr = new StringTokenizer(processed_value);
5     while (itr.hasMoreTokens()) {
6         word.set(itr.nextToken());
7         context.write(word, one);
8     }
9 }
```