MapReduce, Pig, HCatalog and Oozie: Una guía práctica

Luis F. Rivera
Departamento Académico de Tecnologías de Información y Comunicaciones (TICs)
Universidad Icesi
Cali, Colombia
Email: lfrivera@icesi.edu.co

Resumen

My abstract.

Keywords

map reduce, pig, hive, hcatalog, oozie.

INTRODUCCIÓN

Apache Hadoop es un *framework* de código abierto para el almacenamiento y procesamiento de grandes volúmenes de datos [1]. Generalmente, Hadoop es considerado como un ecosistema, en el que habitan, entre otras, herramientas como *Apache Hive*, *Apache Pig*, y *Apache Oozie*, las cuales fueron concebidas con el objetivo de complementar los cuatro elementos principales del core de Hadoop (HDFS, MapReduce, YARN, y Common)[2].

El objetivo principal de este proyecto consiste en analizar un subconjunto de las herramientas pertenecientes al ecosistema de Hadoop, desde la perspectiva de la infraestructura computacional necesaria ponerlas en marcha y de los principales atributos de calidad involucrados en el uso de las mismas. Para este propósito, un escenario de pruebas controlado fue configurado usando la versión 5.10.1 de *Cloudera Manager* (CDH 5.10.1). Dicha configuración fue establecida como se muestra en la figura

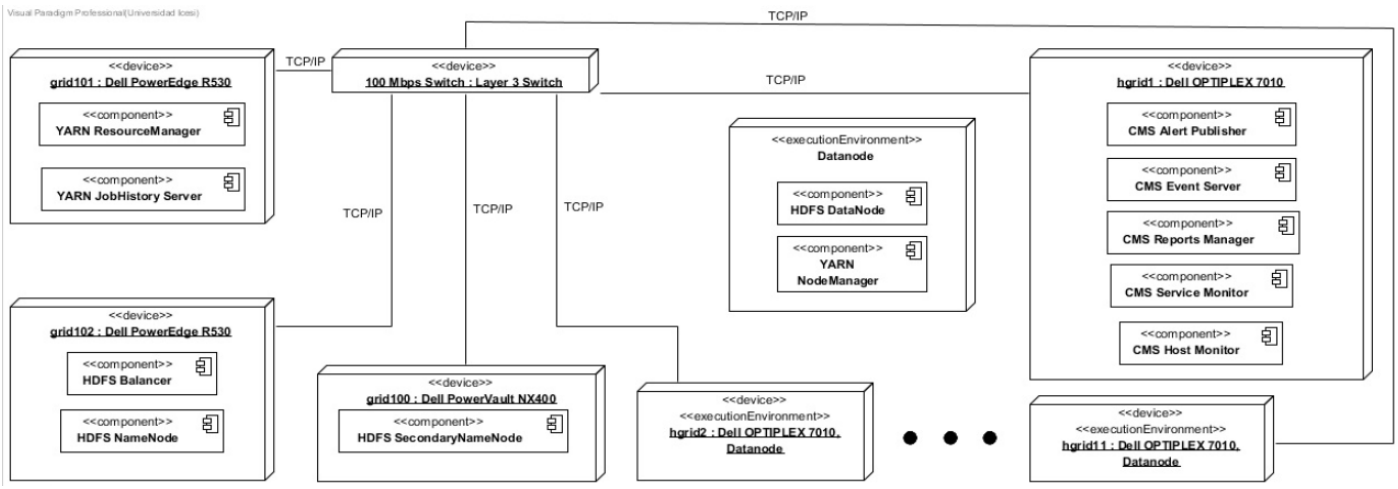

Figura 1. Diagrama de despliegue de la configuración de CDH 5.10.1 en el laboratorio *LIASON* de la Universidad Icesi.

Cada uno de los casos de estudio de minería de datos presente en este proyecto se basa en los datos provistos por el *National Climatic Data Center* (NCDC). Dichos datos son recolectados por medio de sensores climáticos, los cuales recolectan información cada hora, de forma diaria, en distintas estaciones a lo largo del mundo. Dichos datos se encuentran registrados a través de lineas en archivos de texto. La figura muestra la descripción del conjunto de datos mencionado previamente.

[1] https://hortonworks.com/apache/hadoop/
[2] http://www.bmc.com/guides/hadoop-ecosystem.html

```
0057
332130     # USAF weather station identifier
99999      # WBAN weather station identifier
19500101   # observation date
0300       # observation time
4
+51317     # latitude (degrees x 1000)
+028783    # longitude (degrees x 1000)
FM-12
+0171      # elevation (meters)
99999
V020
320        # wind direction (degrees)
1          # quality code
N
0072
1
00450      # sky ceiling height (meters)
1          # quality code
C
N
010000     # visibility distance (meters)
1          # quality code
N
9
-0128      # air temperature (degrees Celsius x 10)
1          # quality code
-0139      # dew point temperature (degrees Celsius x 10)
1          # quality code
10268      # atmospheric pressure (hectopascals x 10)
1          # quality code
```

Figura 2. Descripción del conjunto de datos. Tomado de [1]

El resto del presente documento se encuentra distribuido como se muestra a continuación. En la sección 1 se describen formalmente el objetivo general y los objetivos objetivos específicos del proyecto. En la sección 2 se presentan los tipos de pruebas ejecutadas sobre las herramientas *MapReduce*, *Apache Pig*, y *Apache Hive* para entender la diferencia entre los tiempos de ejecución de las mismas. En la sección 3 se ilustran las distintas pruebas llevadas a cabo sobre las herramientas *Apache Pig* y *HCatalog* para verificar la posibilidad de extender las funcionalidades y capacidades provistas por *Pig*. En la sección 4 se detallan las pruebas realizadas sobre *Apache Ooozie*, las cuales buscan evidenciar la re-usabilidad y mantenibilidad de las aplicaciones desarrolladas sobre esta herramienta. La sección 5 muestra los resultados de la ejecución de las pruebas descritas previamente. En la sección 6 se presentan las conclusiones del presente trabajo. Finalmente, la sección 7 muestra las posibilidades de trabajo futuro que se podrían desarrollar a partir de lo que aquí se presenta.

I. Objetivos del proyecto

El objetivo general del proyecto consiste en analizar, desde un punto de vista arquitectónico, algunas de las herramientas que conforman el ecosistema de Hadoop. Para esto, se estudiará MapReduce, Apache Pig, Apache Hive, HCatalog, y Apache Oozie desde la perspectiva de los atributos de calidad de desempeño (*performance*), reusabilidad (*reusability*), extensibilidad (*extensibility*), y mantenibilidad (*maintainability*)). A continación se describen los objetivos específicos del proyecto y los atributos de calidad asociados a cada uno de estos.

1. Construir y ejecutar un caso de estudio bajo un entorno de pruebas controlado, el cual permita, en una primera instancia, entender la diferencia en los tiempos de ejecución de MapReduce, ApachePig, y Apache Hive. Atributo de calidad relacionado: *performance*.
2. Construir y ejecutar un caso de estudio bajo un entorno de pruebas controlado, el cual permita, en una primera instancia, entender si Apache Pig podría aprovechar las ventajas de Apache Hive a través de HCatalog. Atributos de calidad relacionados: *performance, extensibility*.
3. Construir y ejecutar un caso de estudio bajo un entorno de pruebas controlado, el cual permita, en una primera instancia, entender la posibilidad de re-uso y la facilidad de mantener *workflows* en Apache Oozie. Atributos de calidad relacionados: *reusability, maintainability*.

II. MapReduce, Pig y Hive

En esta sección se presenta la comparación de los tiempos de ejecución de MapReduce, Apache Pig, y Apache Hive para el cálculo de la temperatura máxima registrada por año.

II-A. Ejecución con MapReduce

A continuación se detallan los pasos necesarios para ejecutar el programa *MaxTemperature* en su versión MapReduce-Java.

1) Preparación: Inicialmente, es necesario compilar el código fuente del programa *MaxTemperature* en su versión Java. Para hacer esto, es necesario clonar el repositorio del libro *Hadoop: The Definitive Guide*[3]. Una vez hecho lo anterior, se debe proceder a compilar los archivos fuente necesarios mediante *Maven*.Finalmente, la ruta del archivo .jar compilado deberá establecerse en una variable de entorno llamada *HADOOP_CLASSPATH*.

```
//Clonación del repositorio.
git clone https://github.com/tomwhite/hadoop-book.git

// Compilación del código fuente.
mvn package -DskipTests

// Definición del classpath de Hadoop.
export HADOOP_CLASSPATH=/home/sas6/Oozie-Pig-HCatalog-Demos/assets/hadoop-examples.jar
```

2) Ejecución del programa MaxTemperature: Una vez compilado el código fuente y definida la variable de entorno correspondiente, se procede a ejecutar el programa *MaxTemperature*.

```
hadoop MaxTemperature /user/hive/warehouse/weather_external/full_data.txt out_mr_300GB
```

3) Seguimiento a la ejecución del programa: Una vez iniciada la ejecución del programa, es posible monitorear el progreso del mismo por medio de la consola donde éste se ejecutó o por medio de la interfaz gráfica de YARN.

```
[root@grid102 Oozie-Pig-HCatalog-Demos]# hadoop MaxTemperature /user/hive/warehouse/weather_external/full_data.txt out_mr_300GB
17/07/22 14:23:45 INFO client.RMProxy: Connecting to ResourceManager at grid102.icesi.edu.co/192.168.161.43:8032
17/07/22 14:23:45 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and
 execute your application with ToolRunner to remedy this.
17/07/22 14:23:46 INFO input.FileInputFormat: Total input paths to process : 1
17/07/22 14:23:47 INFO mapreduce.JobSubmitter: number of splits:2314
17/07/22 14:23:47 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1500744570838_0005
17/07/22 14:23:47 INFO impl.YarnClientImpl: Submitted application application_1500744570838_0005
17/07/22 14:23:47 INFO mapreduce.Job: The url to track the job: http://grid102.icesi.edu.co:8088/proxy/application_1500744570838_0005/
17/07/22 14:23:47 INFO mapreduce.Job: Running job: job_1500744570838_0005
17/07/22 14:23:52 INFO mapreduce.Job: Job job_1500744570838_0005 running in uber mode : false
17/07/22 14:23:52 INFO mapreduce.Job:  map 0% reduce 0%
17/07/22 14:24:09 INFO mapreduce.Job:  map 1% reduce 0%
17/07/22 14:24:11 INFO mapreduce.Job:  map 2% reduce 0%
17/07/22 14:24:14 INFO mapreduce.Job:  map 3% reduce 0%
17/07/22 14:24:21 INFO mapreduce.Job:  map 4% reduce 0%
^[[15~17/07/22 14:24:27 INFO mapreduce.Job:  map 5% reduce 0%
```

Figura 3. Monitoreo de la ejecución del programa *MaxTemperature* en MapReduce por medio de la consola en donde éste se ejecutó.

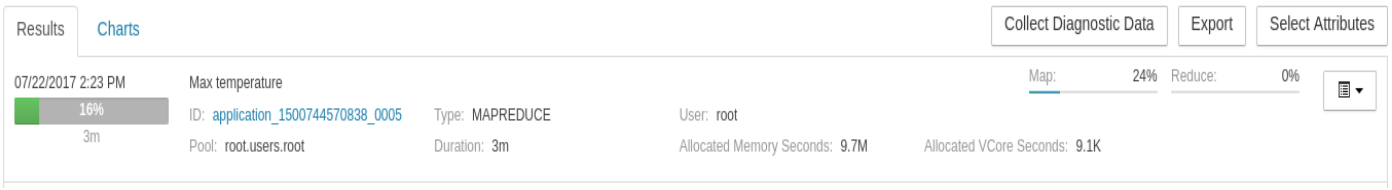

Figura 4. Monitoreo de la ejecución del programa *MaxTemperature* en MapReduce por medio de la interfaz de YARN.

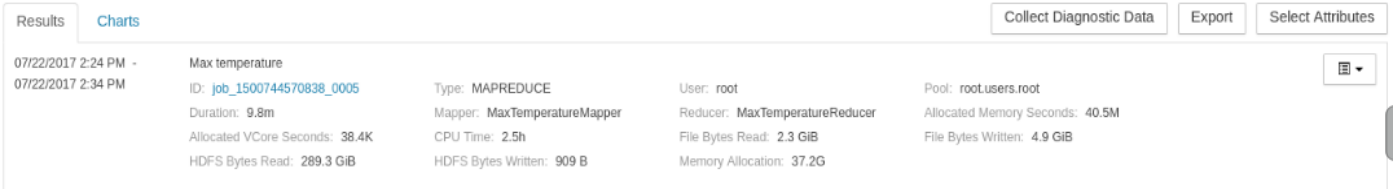

Figura 5. Ejecución finalizada.

[3]Repositorio provisto por Tom White en https://github.com/tomwhite/hadoop-book

II-B. Ejecución con Hive

A continuación se detallan los pasos necesarios para ejecutar el programa *MaxTemperature* en su versión Apache Hive.

1) Ejecución con la consola de Hive: El siguiente script detalla la ejecución del programa *MaxTemperature* en su versión Apache Hive.

```
ADD jar /usr/lib/hive/lib/hive-contrib-1.1.0-cdh5.10.1.jar;
INSERT OVERWRITE DIRECTORY 'out_max_hive_300GB'
SELECT observation_date_year, MAX(air_temperature)
FROM weather_managed
WHERE air_temperature != 9999 AND at_quality_code IN (0,1,4,5,9)
GROUP BY observation_date_year;
```

2) Seguimiento a la ejecución del programa: El monitoreo de la ejecución del programa podrá realizarse a través de la interfaz gráfica de YARN, o por la información proporcionada por el Job History Server.

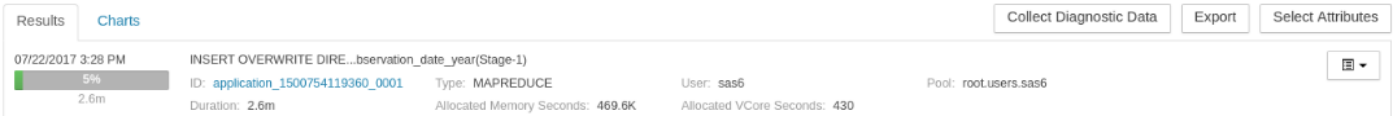

Figura 6. Monitoreo de la ejecución del programa *MaxTemperature* en Hive por medio de la interfaz de YARN.

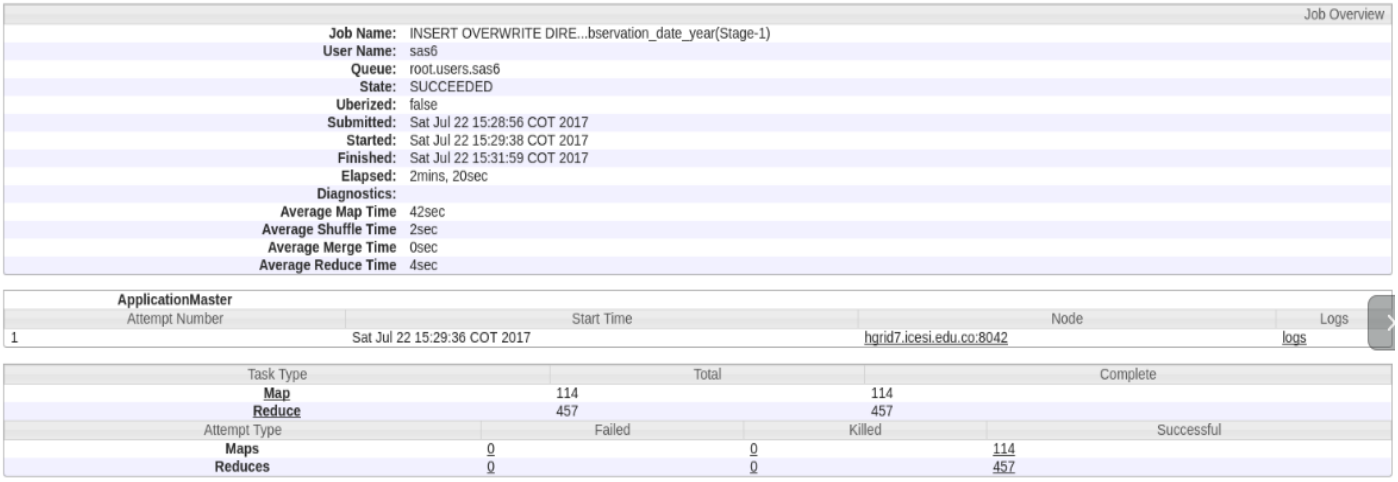

Figura 7. Ejecución finalizada, vista desde el Job History Server.

II-C. Ejecución con Pig

A continuación se detallan los pasos necesarios para ejecutar el programa *MaxTemperature* en su versión Apache Pig.

1) Definición del script en PigLatin: El siguiente script contiene el código utilizado utilizado para ejecutar el programa *MaxTemperature* en su versión PigLating. En las dos primeras lineas del script se detalla el uso de una UDF (*User defined function*), provista por Tom White, para la lectura de registros a partir de la definición de rangos de lectura para sus atributos. El contenido del script fue guardado en un archivo llamado *max-temp.pig*.

```
REGISTER $load_loc;
records = LOAD '$in_s1' USING com.hadoopbook.pig.CutLoadFunc('16-19,88-92,93-93') AS (year:int,
    temperature:int, quality:int);
filtered_records = FILTER records BY temperature != 9999 AND com.hadoopbook.pig.IsGoodQuality(
    quality);
grouped_records = GROUP filtered_records BY year;
max_temp = FOREACH grouped_records GENERATE group,MAX(filtered_records.temperature);
STORE max_temp INTO '$out_max';
```

2) Definición de los parámetros del script: Una vez definido el script en PigLatin, se procede a definir en un nuevo archivo los parámetros necesarios para la correcta ejecución del script. Los parámetros mencionados fueron guardados en un archivo llamado *max.param*.

```
# Load function location.
load_loc=/home/sas6/Oozie-Pig-HCatalog-Demos/assets/pig-examples.jar
# Input.
in_s1=/user/hive/warehouse/weather_external/full_data.txt
# Output.
out_max=out_max_pig
```

3) Ejecución con Grunt en modo batch: A continuación se detalla el comando utilizado para ejecutar el programa *MaxTemperature* mediante el modo *batch* de Grunt.

```
pig -param_file /home/sas6/Oozie-Pig-HCatalog-Demos/scripts/pig/300GB/max.param /home/sas6/Oozie-Pig-HCatalog-Demos/src/pig/max-temp.pig
```

4) Seguimiento a la ejecución del programa: El monitoreo de la ejecución del programa podrá realizarse a través de Grunt, por medio de la interfaz gráfica de YARN, o por la información proporcionada por el Job History Server.

```
2017-07-22 12:31:02,505 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Processing aliases
 filtered_records,grouped_records,max_temp,records
2017-07-22 12:31:02,505 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - detailed locations
: M: records[2,10],records[-1,-1],filtered_records[3,19],max_temp[5,11],grouped_records[4,18] C: max_temp[5,11],grouped_records[4,18] R:
 max_temp[5,11]
2017-07-22 12:31:02,590 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 0% complete
2017-07-22 12:32:09,727 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 4% complete
```

Figura 8. Monitoreo de la ejecución del programa *MaxTemperature* en Pig por medio de la consola Grunt desde donde se ejecutó.

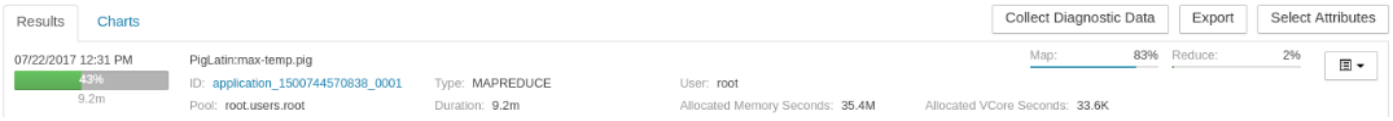

Figura 9. Monitoreo de la ejecución del programa *MaxTemperature* en Pig por medio de la interfaz de YARN.

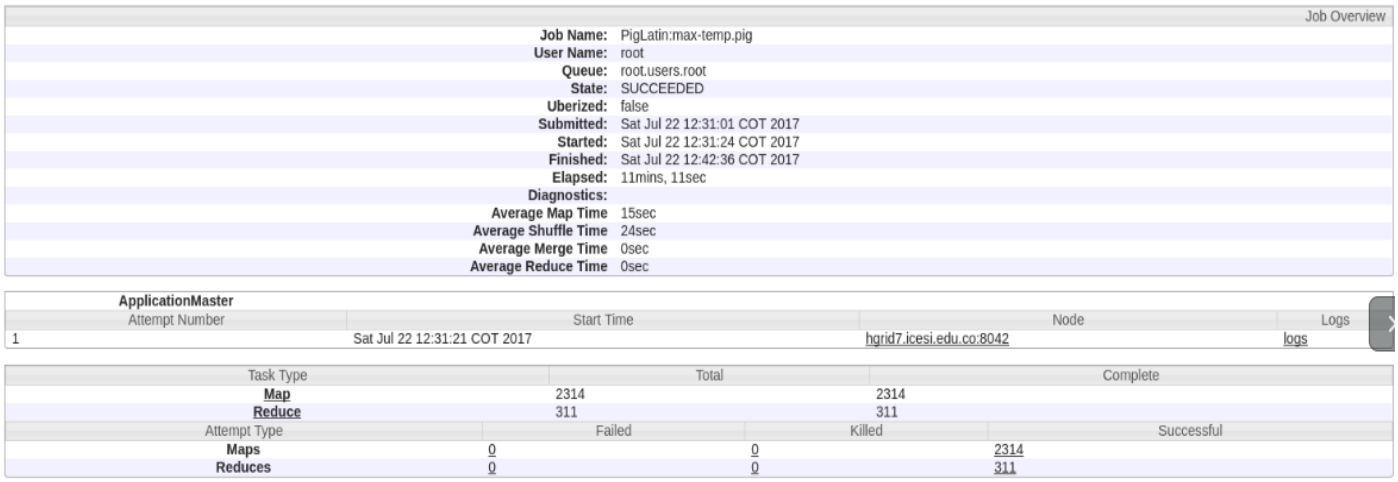

Figura 10. Ejecución finalizada, vista desde el Job History Server.

III. Pig and HCatalog

Pig and HCatalog.

IV. Oozie

Oozie.

V. Results

Results.

VI. Conclusions

Conclusions.

VII. Future work

Future work.

Referencias

[1] T. White, *Hadoop: The Definitive Guide*. O'Reilly Media, Inc., 2012.