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.

I. Introduction

My introduction.

II. Objetivos del proyecto

My objectives.

Question	Condition	Related QAs	Ecosystem Apps	Dataset	Program(s)	Current State
How big is the gap between Pig and MapReduce execution times?	Proposed	Performance	Pig, MapReduce, Oozie	NCDC Weather	Max temperature and Mean maximun temperature station-day-month	Executed on cluster
Can Pig exploit the benefits[1] of Hive through HCatalog?	Proposed	Performance, Extensibility	Pig, HCatalog	NCDC Weather	Partitioned weather	Tested on Cluster
How easily does Oozie support changes in workflow apps?	Proposed	Reusability, Maintainability	Pig, MapReduce, Oozie	NCDC Weather	Mean maximun temperature station-day-month	Tested on Cloudera VM
What is the gap between Oozie and JobControl execution times?	Desired	Performance	Oozie, MapReduce	NCDC Weather	Mean maximun temperature station-day-month	JobControl instance to be coded, tested and executed.

III. 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.

III-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*[2]. 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*.

```
1 //Clonación del repositorio.
2 git clone https://github.com/tomwhite/hadoop-book.git
3
4 // Compilación del código fuente.
5 mvn package -DskipTests
6
7 // Definición del classpath de Hadoop.
8 export HADOOP_CLASSPATH=/home/sas6/Oozie-Pig-HCatalog-Demos/assets/hadoop-examples.jar
```

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

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*.

```
1 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.

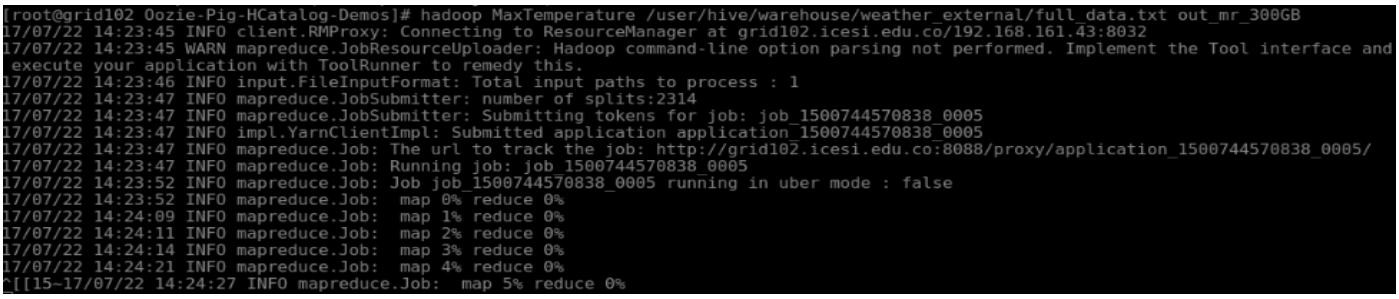

Figura 1. Monitoreo de la ejecución del programa *MaxTemperature* en MapReduce por medio de la consola en donde éste se ejecutó.

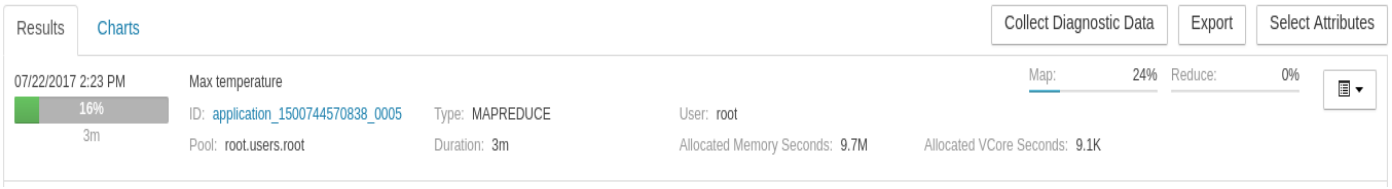

Figura 2. Monitoreo de la ejecución del programa *MaxTemperature* en MapReduce por medio de la interfaz de YARN.

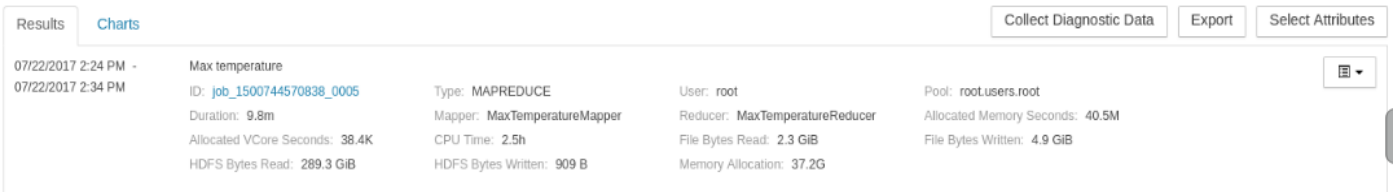

Figura 3. Ejecución finalizada.

III-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.

```
1 ADD jar /usr/lib/hive/lib/hive-contrib-1.1.0-cdh5.10.1.jar;
2 INSERT OVERWRITE DIRECTORY 'out_max_hive_300GB'
3 SELECT observation_date_year, MAX(air_temperature)
4 FROM weather_managed
5 WHERE air_temperature != 9999 AND at_quality_code IN (0,1,4,5,9)
6 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.

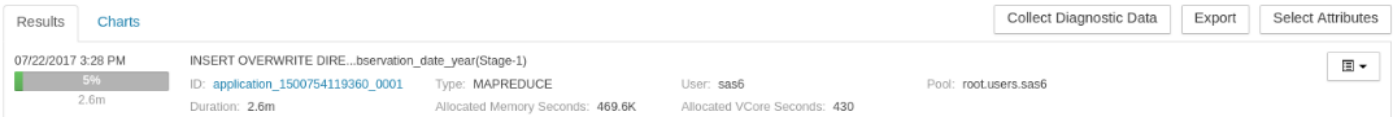

Figura 4. Monitoreo de la ejecución del programa *MaxTemperature* en Hive por medio de la interfaz de YARN.

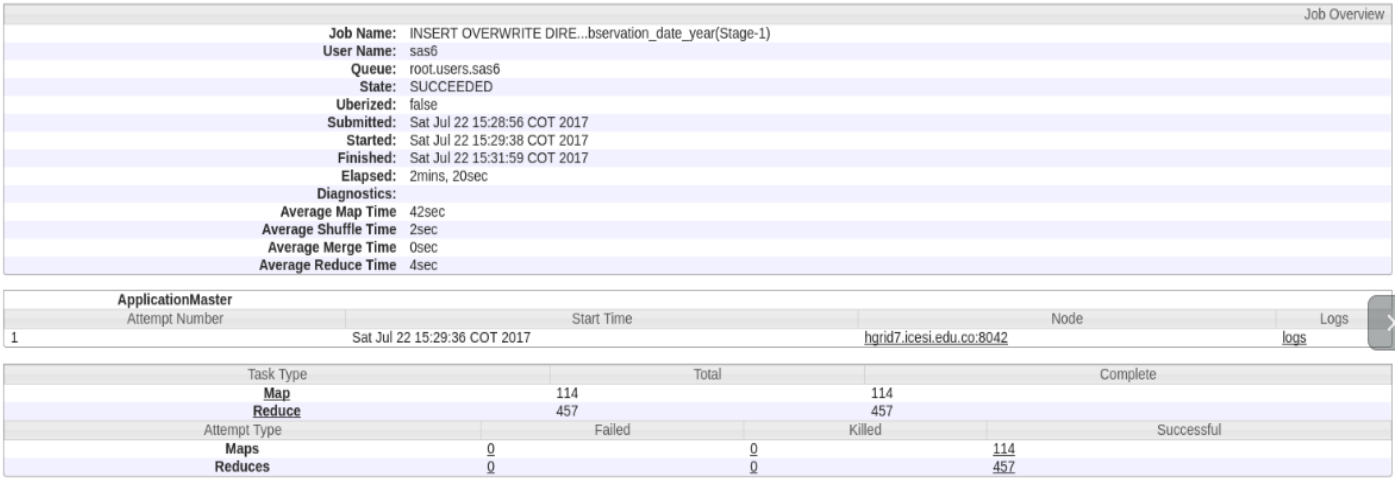

Figura 5. Ejecución finalizada, vista desde el Job History Server.

III-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*.

```
1 REGISTER $load_loc;
2 records = LOAD '$in_s1' USING com.hadoopbook.pig.CutLoadFunc('16-19,88-92,93-93') AS (year:int,
      temperature:int, quality:int);
3 filtered_records = FILTER records BY temperature != 9999 AND com.hadoopbook.pig.IsGoodQuality(
      quality);
4 grouped_records = GROUP filtered_records BY year;
5 max_temp = FOREACH grouped_records GENERATE group,MAX(filtered_records.temperature);
6 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*.

```
1 # Load function location.
2 load_loc=/home/sas6/Oozie-Pig-HCatalog-Demos/assets/pig-examples.jar
3 # Input.
4 in_s1=/user/hive/warehouse/weather_external/full_data.txt
5 # Output.
6 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 6. Monitoreo de la ejecución del programa *MaxTemperature* en Pig por medio de la consola Grunt desde donde se ejecutó.

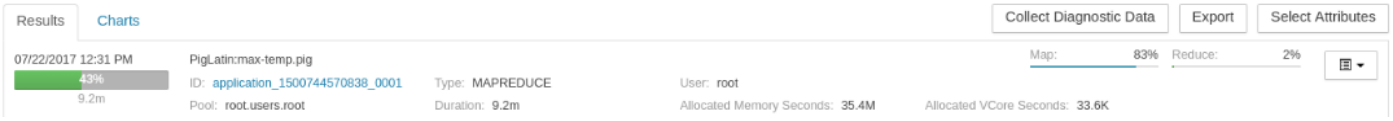

Figura 7. Monitoreo de la ejecución del programa *MaxTemperature* en Pig por medio de la interfaz de YARN.

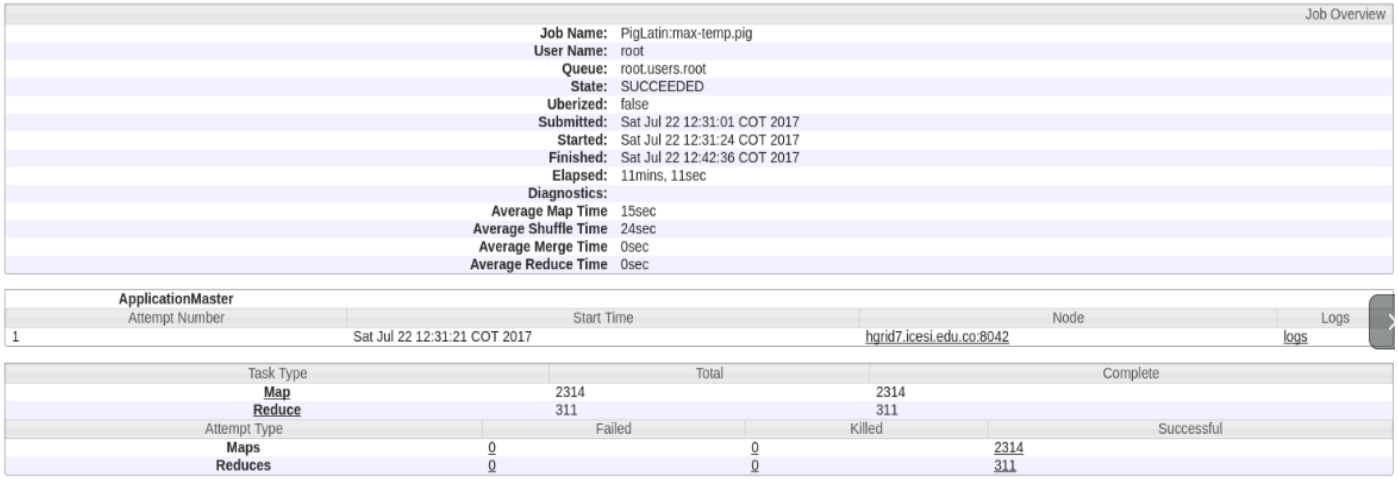

Figura 8. Ejecución finalizada, vista desde el Job History Server.

IV. Pig and HCatalog

Pig and HCatalog.

V. Oozie

Oozie.

VI. Results

Results.

VII. Conclusions

Conclusions.

VIII. Future work

Future work.

Referencias

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.