Big Data

Trabajo Práctico 1

Crespo, Alvaro	50758
Petit, Alejandro	48308
Susnisky, Dario	50592
Videla, Máximo	51071

07 de octubre de 2013

1. Problemas encontrados

Al correr inicialmente los jobs de map-reduce que utilizaban tablas de HBase, tuvimos errores por no setear correctamente la configuración de Zookeeper. Estos errores pudieron solucionarse configurando correctamente las propiedades `HBASE CONFIGURATION ZOOKEEPER QUORUM` y `HBASE CONFIGURATION ZOOKEEPER CLIENTPORT`.

Al momento de implementar joins con los archivos CSV, nos encontramos con el problema de como hacer que los archivos estén disponibles en cada mapper. La clase *Distributed Cache* nos sirvió para justamente sobrepasar esta dificultad.

Al correr Pig en forma map-reduce con hadoop pseudo-distribuido, tuvimos problemas en un momento ya que hadoop recordaba la IP externa de la computadora y, por lo tanto, no encontraba el reducer.

En la métrica 10 de Pig, tuvimos algunos inconvenientes para hacer el join ya que, en algunos casos, no había vuelos cancelados y, en otros, no había vuelos que hayan despegado, por lo tanto, quedaban campos en null. Se solucionó haciendo un FULL OUTER JOIN y chequeando que los campos fueran null y reemplazando esos valores por 0. También se tomaban los valores de las fechas que no fueran null.

En varios momentos, tanto al implementar las métricas con MapReduce en Java y con Hive, nos encontramos con irritantes errores de formato en algunos archivos CSV. En particular, en los archivos *airports.csv* y *carriers.csv*, todos los campos estabán entre comillas dobles (“) y esto producía algunos errores al hacer joins. Es por esto que previo a hacer los joins tuvimos que efectuar algunas moficiaciones en los datos, obviamente sin modificar los archivos del HDFS. Simplemente es eliminaron las comillas dobles previo a la utilización de los datos de estos archivos.

Respecto a Hive, encontramos ciertos problemas al parametrizar los *scripts*. Esto trajo particulares complicaciones al implementar la métrica 6, ya que en dicha métrica es posible recibir o no un parametro. En nuestros *scripts* explotamos el hecho de que Hive permite el uso de variables del sistema mediante la linea de comandos (con el uso de -hiveconf). Una vez establecidas estas variables, Hive reemplaza las apariciones de estas variables por su valor real de forma similar a como funciona una macro. Sin embargo, en caso de no existir alguna variable, Hive las toma como texto plano. Para poder manejar estas situaciones, creamos una UDF que trata de discriminar si la variable existe, abusando del conocimiento que toda variable no existente contenera ̈hiveconfën su texto.

Un segundo problema con Hive, a la hora de imprimir las salidas a archivos. La separación de los campos y las tuplas no son cómodos a la hora de leerlos, haciendo las salidas poco legibles. Esto podría solucionarse rápidamente con un *script* de *Bash* o ejecutando busquedas y reemplazos en editores con esta capacidad (como por ejemplo, *vim*).

Además, cabe mencionar que la métrica 5, tiene un elevado tiempo de ejecución. Este es un problema de especial interés ya que al ejecutarlo en sistemas locales, esto no se presentaba como un problema y fue evidente al probarlo en el *cluster*. Sería una posible mejora optimizar esta métrica para que su tiempo de ejecución sea menor al actual.

Por último, nos encontramos con un problema al crear y popular las tablas medianto

los scripts en Hive. Al correr los scripts, los archivos csv originales se veian borrados. Finalmente logramos sortear este problema definiendo tablas externas y modificando la manera en la que son populadas.

Con respecto a los unit test, se instaló Cobertura como plugin de eclipse para verificar su funcionamiento, y luego se procedió a agregar como plugin en el pom. Al momento de compilar con mvn, este se encarga de generar los reportes correspondientes en XML y HTML. Se agrego también al pom las dependencias de JUnit para testeo unitario y MRUnit para testear las implementaciones de Map-Reduce. La API de MRUnit permite realizar los tests sobre los métodos de map y reduce. Es decir que brinda una interfaz para facilitar la verificación del correcto funcionamiento de los mismos. MRUnit permite crear y enviar una entrada al mapper y al reducer y volcar su salida de forma tal de poder compararla con una salida esperada y conocida.

2. Decisiones de implementación

Para las métricas implementadas con MapReduce en Java, decidimos implementar, en todos los casos, *Broadcast Join* al hacer joins por 2 razones principales: en todos los casos, uno de los *dataset* siempre se podía asumir pequeño (3000 aeropuertos, 1500 aerolíneas y 5000 aviones), y además es la opción más fácil de implementar (en algunos casos se utilizaban tablas de *HBase* y en otros se aprovechaba el *Distributed Cache* de Hadoop para distribuir los archivos CSV).

Decidimos implementar 2 simples métricas extra, como son la cantidad de vuelos totales de cada aerolínea, y la proporción de vuelos cancelados sobre el total de vuelos para cada aerolínea. Otras métricas interesantes hubieran sido agregarle a estas métricas extra la posibilidad de discriminar por año además de por aerolínea, ya sea la cantidad de vuelos o la proporción de vuelos cancelados.

En el caso de las métricas de *Pig* tomamos las siguientes decisiones a la hora de implementarlas:

En la métrica 9, se cuentan tanto los vuelos cancelados como los que salieron porque la ruta, a nuestro entender, no depende de si sale o no el vuelo sino, de los vuelos programados para salir.

En la métrica 11, para que la hora figure con los ‘:’ en el medio lo separamos en dos int, uno para horas y otro para minutos, el único problema de esto es que las ‘6:02’ las representamos como ‘6:2’ pero no consideramos que sea un detalle importante. En el caso de que un aeropuerto no tenga despegues, se ve el nombre del aeropuerto y se deja vacío el campo de la hora, de esta forma, no se pierde la información de que no hubo vuelos ese día.

En la métrica 12, no tomamos en cuenta los casos en que el despegue real ocurra antes que el programado, ya que la “demora” en este caso sería negativa y no tendría sentido. Además, siempre que el vuelo salga a la hora esperada o antes, se puede decir que no hay demora, es decir, la demora es 0.

3. Instrucciones para ejecutar las métricas

3.1. Map Reduce

3.1.1. Métrica 1 - Promedio de demora de despegue por mes por estado

hadoop jar bigdata-tp1-jar-with-dependencies.jar -inPath 'input_path' -outPath 'output_path' -avgTakeOffDelay

3.1.2. Métrica 2 - Vuelos Cancelados por aerolínea

hadoop jar bigdata-tp1-jar-with-dependencies.jar -inPath 'input_path' -outPath 'output_path' -cancelledFlights -carriersPath 'carriers_path'

3.1.3. Métrica 3 - Millas voladas por aerolínea por año

hadoop jar bigdata-tp1-jar-with-dependencies.jar -inPath 'input_path' -outPath 'output_path' -milesFlown -carriersPath 'carriers_path'

3.1.4. Métrica 4 - Horas de vuelo por fabricante

hadoop jar bigdata-tp1-jar-with-dependencies.jar -inPath 'input_path' -outPath 'output_path' -flightHours -manufacturer 'target_manufacturer_name'

3.1.5. Métrica 13(OPCIONAL) - Cantidad de vuelos por aerolínea

hadoop jar bigdata-tp1-jar-with-dependencies.jar -inPath 'input_path' -outPath 'output_path' -flightCount -carriersPath 'carriers_path'

3.1.6. Métrica 14(OPCIONAL) - Proporción de vuelos cancelados por aerolínea

hadoop jar bigdata-tp1-jar-with-dependencies.jar -inPath 'input_path' -outPath 'output_path' -propCancelledFLights -carriersPath 'carriers_path'

3.2. Hive

Es importante mencionar que para los archivos de entrada, Hive recibe directorios y no archivos puntuales. Por esto, es importante que los directorios únicamente contengan aquellos archivos que deseamos que sean procesados. En el caso de los *script* que reciben información de aeropuertos, airports.csv debe estar en un directorio de manera aislada. Por último, es importante aclarar que los directorios de los archivos de entrada deben ser *paths* absolutos y no relativos.

3.2.1. Métrica 5 - Top 5 Aeropuertos con demora de despegue por año

hive -S -f metric5-depDelayTop5.sql -hiveconf flightsPath='input_flights_path' -hiveconf airportsPath='input_airports_path' -hiveconf output='output_path'

3.2.2. Métrica 6 - Imprevistos 2005

hive -S -f metric6-2005FlightStats.sql -hiveconf flightsPath='input_flights_path' [-hiveconf airport='airport_IATA'] -hiveconf output='output_path'

3.2.3. Métrica 7 - Top 5 Aeropuertos con mayor promedio de demoras

hive -S -f metric7.sql -hiveconf flightsPath='input_flights_path' -hiveconf airportsPath='input_airports_path' -hiveconf output='output_path'

3.2.4. Métrica 8 - Huracanes con más cancelaciones

hive -S -f metric8.sql -hiveconf flightsPath='input_flights_path' -hiveconf output='output_path'

3.3. Pig

3.3.1. Métrica 9 - Rutas más voladas por año

pig -param flights=FLIGHTS_PATH -param airports=AIRPORTS_PATH -param output=OUTPUT_PATH ej9.pig

3.3.2. Métrica 10 - Cantidad de vuelos cancelados y no cancelados en septiembre de 2011

pig -param flights=FLIGHTS_PATH -param output=OUTPUT_PATH ej10.pig

3.3.3. Métrica 11 - Hora partida del último vuelo del 9/11 para cada aeropuerto

pig -param flights=FLIGHTS_PATH -param output=OUTPUT_PATH ej11.pig

3.3.4. Métrica 12 - Promedio diario de demora de despegue en el año 2001

pig -param flights=FLIGHTS_PATH -param output=OUTPUT_PATH ej12.pig

4. Formato de los resultados de las métricas

4.1. Map Reduce

4.1.1. Métrica 1 - Promedio de demora de despegue por mes por estado

Los resultados de esta métrica tienen el siguiente formato:

ESTADO-MES PROMEDIO

donde el estado se representa por dos letrás mayúsculas (su código postal) y el mes escrito en letras y en ingleś.

Un ejemplo sería

WY-September 10.518987341772151

4.1.2. Métrica 2 - Vuelos Cancelados por aerolínea

Los resultados de esta métrica tienen el siguiente formato:

AEROLINEA CANTIDAD

Un ejemplo sería

United Air Lines Inc. 34

4.1.3. Métrica 3 - Millas voladas por aerolínea por año

Los resultados de esta métrica tienen el siguiente formato:

AEROLINEA-AÑO CANTIDAD

Un ejemplo sería

Delta Air Lines Inc.-1987 1171792

4.1.4. Métrica 4 - Horas de vuelo por fabricante

Los resultados de esta métrica tienen el siguiente formato:

NRO_AVION CANTIDAD_HORAS

Cabe destacar que la cantidad de horas se encuentra en decimal ya que no se efectuaron redondeos.

Un ejemplo sería

N997AT 191.93333333333334

4.1.5. Métrica 13(OPCIONAL) - Cantidad de vuelos por aerolínea

Los resultados de esta métrica tienen el siguiente formato:

AEROLINEA CANTIDAD

Un ejemplo sería

United Air Lines Inc. 324

4.1.6. Métrica 14(OPCIONAL) - Proporción de vuelos cancelados por aerolínea

Los resultados de esta métrica tienen el siguiente formato:

AEROLINEA PROPORCION

Un ejemplo sería

Delta Air Lines Inc. 0.008937960042060988

4.2. Hive

El output de Hive se puede ver en un archivo en el directorio especificado al ejecutar el *script*. Como fue mencionado en los problemas encontrados, la separación entre los campos se da por el caracter ^A mientras que las tuplas están separados por \N. A continuación se detalla el órden de los campos para cada una de las métricas.

4.2.1. Métrica 5 - Top 5 Aeropuertos con demora de despegue por año

Los resultados de esta métrica tienen el siguiente formato:

AÑO - PUESTO - AEROPUERTO - HORAS TOTALES DE DEMORA

Un ejemplo sería

2000 - 1 - John F Kennedy Intl - 456.23

4.2.2. Métrica 6 - Imprevistos 2005

Los resultados de esta métrica tienen el siguiente formato:

FECHA - CANTIDAD DE VUELOS DEMORADOS - SUMA DE HORAS DE DEMORA - CANTIDAD DE VUELOS DEMORADOS - CANTIDAD DE VUELOS DESVIADOS - CANTIDAD DE VUELOS CANCELADOS POR MAL CLIMA

Un ejemplo sería

25/04/2005 - 123 - 456 - 789 - 58 - 123

4.2.3. Métrica 7 - Top 5 Aeropuertos con mayor promedio de demoras

Los resultados de esta métrica tienen el siguiente formato:

AEROPUERTO - PUESTO - PROMEDIO DE VUELOS DEMORADOS POR DIA

Un ejemplo sería

John F Kennedy Intl - 1 - 89.53

4.2.4. Métrica 8 - Huracanes con más cancelaciones

Los resultados de esta métrica tienen el siguiente formato:

HURACAN - FECHA - CANTIDAD DE VUELOS CANCELADOS

Un ejemplo sería

KATRINA - 25/04/2005 - 123

4.3. Pig

4.3.1. Métrica 9 - Las rutas más voladas

Los resultados de esta métrica tienen el siguiente formato:

AÑO ORIGEN DESTINO CANTIDAD_VUELOS

Se ordenan por año y, dentro de cada año, por cantidad de vuelos. Sólo se toman en cuenta los primeros diez de cada año.

Un ejemplo sería

2000 Warsaw Municipal Salem Memorial 123456

4.3.2. Métrica 10 - Cantidad de vuelos cancelados y no cancelados en septiembre de 2011

Los resultados de esta métrica tienen el siguiente formato:

FECHA CANTIDAD_VUELOS CANTIDAD_CANCELADOS

Cabe destacar que no se tiene en cuenta los casos de los vuelos reprogramados, es decir, un vuelo cancelado que se efectúa otro día en el mismo mes, suma a ambos.

Un ejemplo sería

29/9/2001 112 14

4.3.3. Métrica 11 - Hora partida del último vuelo del 9/11 para cada aeropuerto

Los resultados de esta métrica tienen el siguiente formato:

ORIGEN HORARIO

Cabe destacar que las horas se muestran en el formato h:m pero no se agrega un 0 adelante de los minutos si son menores a 10, es decir, las seis y cinco de la tarde se representan de la siguiente manera: 6:5. Otro comentario sería que se listan todos los aeropuertos, incluso en los que no se realizó ningún vuelo, estos se indican dejando la hora en blanco.

Un ejemplo sería

SYR 8:46

4.3.4. Métrica 12 - Promedio diario de demora de despegue en el año 2001

Los resultados de esta métrica tienen el siguiente formato:

AÑO-MES-DIA PROMEDIO

Cabe destacar que la cantidad de horas se encuentra en decimal ya que no se efectuaron redondeos.

Un ejemplo sería

2001-12-31 0.298048048048048