CIS 4930 Introduction to Hadoop and Big Data

Secondary Sort Lab

Julie Reyes | U76631122

Lab Introduction:

This lab explores running a MapReduce job under different parameters to determine the overall effects on the output. Each new parameter added to the job will produce a different output to the Hadoop cluster. The different outputs will be analyzed to determine the significance of the reducer, comparator, partitioner, group comparators, and mapper. The purpose of this lab is to develop a more in depth understanding of how to implement a MapReduce job that uses secondary sorting.

Lab subsection 1:

The initial run for this lab included running the job as a Map-only job to visualize the output emitted from the mapper to the reducer. The following is a brief snapshot of the output emitted from the mapper initially:

Map-Only:

```
(Brown,1922)      Brown John 1922-May-29
(Andrews,1935)    Andrews Julie 1935-Oct-01
(Jones,1945)      Jones Zeke 1945-Dec-30
```

Next, assigning two reducer tasks rather than zero through command line parameters, the MapReduce job was performed again. The following depicts some of the results that were produced from each reducer:

Reducer 0:

```
(Addams,1989)     Addams Pugsley 1989-Oct-31
(Bacon,2004)      Bacon Francis 2004-Sep-25
(Bhatt,2008)      Bhatt Barton 2008-Jan-08
```

Reducer 1:

```
(Addams,1920)     Addams Morticia 1920-Apr-30
(Addams,1960)     Addams Jane 1960-Sep-06
(Addams,1964)     Addams Gomez 1964-Sep-18
```

The output from these two reducers was a result of running the MapReduce job with the default partitioner and comparators. Since the reducer is not specified in the driver code or command line the description of the reducer is just the default Hadoop reducer, which is the identity reducer. This default reducer outputs in sorted order, yet there may by the same keys in different reducer files, for example Addams. This is due to not utilizing custom partitioners and comparators.

Lab subsection 2:

A custom partitioner, named NameYearPartitioner, takes in the name and year compound keys emitted from the mapper and uses the last name portion of the key to determine which reducer to send the key value pairs to. The partitioner uses the getLeft() method to select the last name portion of the key and the hashcode() method to return a hashcode for the string. Next, the hashcode generated from the last name is bitwise AND with the max value integer. This number is then MOD with the total number of reducers and the result culminates in the keys with the same last name being sent to the same reducer. The following depicts a portion of the results obtained from running the MapReduce job implementing the custom partitioner:

Reducer 0:

```
(Addams,1920)    Addams Morticia 1920-Apr-30
(Addams,1960)    Addams Jane 1960-Sep-06
(Addams,1964)    Addams Gomez 1964-Sep-18
```

Reducer 1:

```
(Antilla,1936)   Antilla   Lorenzo 1936-Apr-16
(Bacon,1888)       Bacon Kevin 1888-Nov-21
(Bacon,2004)       Bacon Francis 2004-Sep-25
```

From the results above, it is evident that after implementing the custom partitioner, the reducer files do not contain a shared last name between the two. Therefore, the partitioner functions as sorting the data to specific reducers. For example, in the default partitioners there are Addams values in both the reducer files, whereas in the custom partitioner output the Addams values are only in one reducer file.

Lab subsection 3:

The NameYearComparator class uses the name and year pair keys emitted from the mapper and first compares the names. If the names are equal the sort comparator compares the year, where later years are considered as lesser than earlier years. The following represents a small portion of the results obtained when running the MapReduce job with the sort comparator:

Reducer 0:

```
(Addams,1989)     Addams Pugsley 1989-Oct-31
(Addams,1964)     Addams Gomez 1964-Sep-18
(Addams,1960)     Addams Jane 1960-Sep-06
```

Reducer 1:

```
(Antilla,1936)   Antilla   Lorenzo 1936-Apr-16
(Bacon,2004)       Bacon Francis 2004-Sep-25
(Bacon,1888)       Bacon Kevin 1888-Nov-21
```

The final output from this job different from the job without the custom comparator in that the keys are sorted based on last name and date. The same last name key is sorted by descending date, which is different from the previous output. The NameYearComparator functions by extending the WritableComparator. The compare() method overrides the default comparison method to compare the last name first, if the names are equal then the year is compared. Another compare method overrides the default compare method, which uses byte comparisons and is not optimized for the compound key being emitted from the mapper. This method reads the incoming bytes to deserialize the StringPairWritable objects being compared.

Lab subsection 4:

The NameYearReducer extends the reducer class and takes in four parameters of type StringPairWritable and Text. The reduce method overrides the Reducer and takes in three parameters including StringPairWritable key, Iterable<text> values, and Context context. In the reduce method for each value in the set of values emitted from the mapper, only emit the first of the value list. The following is the result of running the MapReduce job with this Reducer implemented:

Reducer 0

```
    Addams Pugsley 1989-Oct-31
    Addams Gomez 1964-Sep-18
    Addams Jane 1960-Sep-06
```

Reducer 1

```
    Antilla   Lorenzo 1936-Apr-16
    Bacon Francis 2004-Sep-25
    Bacon Kevin 1888-Nov-21
```

Lab subsection 5:

The NameComparator class is the custom group comparator class for this lab. This group comparator class functions by comparing only the last name field and disregarding the year field. In this class, pairs that have the same name will be categorized into the same Reducer**.** The group comparator extends the WritableComparator. The compare method compares the name portion of the key only. The following demonstrates the results of the running the MapReduce job with the group comparator implemented:

File 1

```
Addams Pugsley 1989-Oct-31
Andrews Andrew 1944-Apr-04
Bisignano Bronwyn    1963-Nov-04
```

File 2

```
Antilla   Lorenzo 1936-Apr-16
Bacon Francis 2004-Sep-25
Bhatt Barton 2008-Jan-08
```

After running the job with the group comparator, the output changes from the previous run by only showing a single record for a last name. Therefore, the final output of this job is the youngest person with a unique last name.