CSCI - B649 [Spring 2015]

Project 1: WordCount in Hadoop

pracshah@umail.iu.edu, abhivija@indiana.edu, ppkapoor@indiana.edu

The Word Count driver program uses a simple *MapReduce* process to determine the count of the words present in a given input text file. When a *Combiner* is added to this process, it reduces the number of records to be copied from node to node, so that the *Reducer* can work on less number of data records, thereby enabling optimization.

Transformation of data during the computations, i.e. data type of key, value:

The default Hadoop framework Combiner for the WordCount program has an implementation similar to the default Hadoop framework's Reducer. The input in the current program is stored as a text file (chunks of data) and this data is fed to the Mapper class. The *Mapper* processes all the input data and produces <*Key, Value*> pairs as output. The Mapper utilizes *StringTokenizer* class to tokenize the text file, line by line. Once the string is tokenized, the mapper iterates through the tokenized string and generates <Key, Value> pairs. The Mappers output *(Word, 1)* pairs which are then forwarded to the Combiner. For example, a line of text in a given file, which is 'My name is is Prachi', the output of the combiner in the <Key, Value> pair form will be:
My, 1
name, 1
is, 1
is, 1
Prachi, 1

Even though the word 'my' appears twice, the Mapper will consider each word individually and the value for each key will be '1'. Imagine for a huge text file, the number of word occurrences will be multiple and the reducer might need to process several number of records, in this case 5. However, in the case of huge data, a *Combiner* can helps reduce the number of records processed by the Reducer. A Combiner in such a case will take the above 5 records as input and locally perform aggregation of the data. The output of the Combiner will then be:
My, 1
name, 1
is, 2
Prachi, 1

The Combiner aggregates these pairs for the same word occurrence and produces a *(Word, n)* pair records where, n is the number of times the particular word has repeated in the text document. This data is now sent for shuffling.

The *Shuffle and Sort* process happens wherein all the word occurrences from several lines of text respectively are noted, and distributed across several Reducers. The output of this process might be:
My, [1]
name, [1, 3, 5]
is, [2, 4, 6]
Prachi, [1, 1]
…

The *Reducer* then performs further aggregation and adds up all the word occurrences for each of the <Key, Value> pair records and produces the final output which can be as follows:
My, [1]
name, [9]
is, [12]
Prachi, [2]
…

As we can see, the Reducer receives a limited number of records (<Key, Value> pairs) for processing.

Data flow through disk and memory during computations:

The input file in first loaded from the local disk into the Hadoop Distributed File System (HDFS). The given input text file is distributed evenly among all the nodes of the file systems. During the different stages of the data flow, the data is handled as in the (Key, Value) pair format.

The Mapper first performs its operations on the data stored on the HDFS and emits (Word, 1) pair records. Next, the data is passed to the Combiner which locally aggregates these records to produce (Word, n) pair records for each repeated occurrence of each word, respectively.

This data is now sent to the Shuffle and Sort module that shuffles the data to sort it according to the key (word), the process if performed within the Hadoop Framework.

Next, the data is fed to the Reducer which performs the computation on the HDFS and stores the output on the HDFS.

We must note that the Mapper, Combiner, Shuffle and Reducer functions are performed on several nodes on the HDFS.

Reference:

http://developer.yahoo.com/hadoop/tutorial/module4.html
https://www.youtube.com/watch?v=HFplUBeBhcM
https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html
https://www.youtube.com/watch?v=7FcMhTTG1Cs