CEG-7380

Cloud Computing

Spring 2016

Project #4

Report File

DHRUVKUMAR NAVINCHANDRA PATEL

U00791652

K-mer Counting in MapReduce

Following are the different steps which I followed during development of this project

1) **Analysis:** Here I implement K-mer is a substring of length (K>0) counting the occurrence of all substring is a central step in many analysis of DNA sequence data. Counting K-mers for DNA sequence means finding frequencies of K-mers for the entire sequence. Which is used in bioinformatics application. So, here finds the TOP N K-mers for a given N>0.

    1) **Hadoop MapReduce implementation :**

        i) **Mapper Implementation:** In Mapper I am taking data from ecoli.fa file. Which contains long DNA sequence in FASTA format. The input file should contain the following format: It begins with a single –line description (starting “>”) and follows by the lines of sequence data.

        ```
        >gi|49175990|ref|NC_000913.2| Escherichia coli K12 substr. MG1655,
        complete genome
        AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGA
        TTAAAAAAAGAGTGT CTGATAGCAGC
        TTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTATTGA
        CTTAGGTCACTAAAT ACTTTAACCAA
        ```

        a) In mapper generate key as Text and value as IntWritable.

        b) Here I find the 10-mers or 20-mers which is indicating in Driver class from whole long DNA sequence from ecoli.fa input file.

        c) So Output from mapper is mers and put 1 as value for each and every mers which is generated from file.

        ii) **Reducer implementation:** In reducer I am taking the key and value from mapper and generate output in key as Text and value as Intwritable.

        a) In reducer I just calculate total frequency for each key (mers) and store into array list.

        b) I created an array list with 10 size and every time if the size is 10 then sort the array list in ascending order, compare new mers frequency with lowest value in array list and if the new is larger than older than just replace new value with lowest value. So after completing whole iterations I will get final TOP 10 most frequently occurring 10-mers or 20-mers in my array list.

        c) Finally I used protected void cleanup() method which is called at last so I just retrieve the data from my final array list and write into the file in following output format. You can find for both 10-mers and 20-mers output in my output folder which I includes those files in my project.

Output from Virtual Cloud:

```
[Mers] [Frequency]

CGCATCCGGC        150
```

iii) **Driver class –** It contains all the required parameter to execute Map-Reduce Program. Here we have to set explicitly in Configuration parameter about noofmers. For example if you want to find TOP 10 for 20-mers then just change conf.set(“Noof-mers”,”10”); in my driver class change 10 with 20 or whatever you want to find.