SUMMARY

The algorithm is divided into two python files mapper.py and reducer.py.

In the mapper.py python file, it reads the input Posts.xml file line by line. At the beginning of the loop, we create an empty list “index”. In the first for loop, we are appending the indices of double inverted commas to extract each element. Then in the next for loop, we are taking a pair of indices and checking that whether slicing the line in that range of index contains “CreationDate” or not, if it contains “CreationDate” then we will select the next two pairs of indices to extract the value of “CreationDate”. For piping between mapper and reducer, we had printed the key and values pairs separated by single tabs, before doing this we removed the unnecessary characters from the key using the strip command and then split the creationdate by “:” so as to extract year-month-date-hour. here the key is year-month-date-hour and values are 1, e.g. “2017-07-28T06 1”.

In the reducer.py file, we used sys module and prettytable module to plot the table in a systematic order. Before reading the outputs from the mapper we will initialize some variables maxi and mini as 0 and 64851(total no of posts), and created two empty lists “hours” and “years”, one to hold the total no of posts in each hour and another for the total no of posts in each year. Then we will read the lines and split them based on the tab and store the values in the word, and count variables. And on the following code, we stored the total number of posts in each year, each hour in those lists, and each hour of a day appended to the table. In the end, we created two tables with columns “hours”, ”count” and “year”, ”count” plotted the table.

Yes, there is an opportunity for Cloud deployment for Stack Exchange whenever the peak to lowest ratio is greater. We can see the peak to lowest ratio in the output for three different tables.