Project REPORT- COMP9313

(Swati Bharti – z5277828)

**Algorithm Candidate Generation (As implemented in Spark)**

A while loop is considered which runs continuously and only terminates when both the threshold of number of collisions (**αm**) and minimum number of candidates(**βn**) conditions are satisfied.

**Step1:**

**Map** transformation is used on the data\_hases data value(**n**) and then the hash value(**m**) of the data\_hashes is compared with the Query\_hashes and if they collide and the value is greater than the threshold of number of collisions(**αm**) then the data value(**n**) corresponding to the data hash (**m**) is added to the candidate counter. If there is no collison then **None** value is given to such candidates.

**Filter** is used to filter the None values from the obtained result. The filter removes all the values with None value and returns only the data value that has satisfied the threshold number of collisions (**αm**).

After adding the data values to the candidate counter they are counted and total number of candidates in the counter is calculated.

**Step2:**

The total number of candidates is compared with the minimum number of candidates( **βn**)and if the total number is greater than then the while loop breaks, returns the data values stored in the candidate counter and then the algorithm terminates.

Else the offset is incremented by one and once again Step1 repeats until the condition is satisfied.

**Count function( As implemented)**

This function is used to check whether the collision has taken place or not.

The count function takes data\_hashes(**n\*m**),query\_hashes(**m**) and offset as arguments.

A for loop is used to traverse over all the hash values(m) from data\_hashes and query\_hashes.

**zip** function is used to unfold all the tuples at once or else it is difficult to unfold many tuples.

For each value absolute difference between the hash value of data\_hashes and query\_hashes is considered.If the difference is less than or equal to the offset then the counter is incremented by one(which is initially set to zero), else counter is untouched.

After the for loop terminates, the value of the counter is returned to the candidate generation algorithm and its value is further compared with the threshold of number of collisions(**αm**).

**Results:**

The results are given as follows :

Figure1: For the Toy dataset

A screenshot of a cell phone

Description automatically generated

Figure2: For Custom dataset 1

A screenshot of a cell phone

Description automatically generated

Figure3: For Custom dataset 3

A screenshot of a cell phone

Description automatically generated

Figure4: For Custom dataset 4

A screenshot of a social media post

Description automatically generated

Figure5: For Custom dataset 5

**A screenshot of a cell phone

Description automatically generatedPerformance**

🡪For custom dataset 4 and toy dataset ,the performance is almost same,but for custom dataset 5 and custom dataset 1 the performance is not as expected and it is slower compared to other datasets.

🡪Custom dataset 3 performace is great which means that if the number of candidates in candidate set is too less or too high the performance boosts.