Big Data Computing - Homework 1

Group 45 - Gaio Giovanni, Grigolato Giordano, Moretti Simone

We implemented the function **MRPrintStatistics** as follows:

```python
def point_count(list):
    clusters_dict = {}
    for c_id, category in list[1]:
        count = np.array([0,0])
        if category == 'A': count[0] = 1
        if category == 'B': count[1] = 1
        if c_id in clusters_dict: clusters_dict[c_id] += count
        else: clusters_dict[c_id] = count
    return clusters_dict.items()

def MRPrintStatistics(U, C):
    N = U.count()
    L = int(np.sqrt(N))
    triplets = (U.map(lambda x: (np.random.randint(0, L-1), (np.argmin([np.
                                          square(np.linalg.norm(np.array(x[0
                                          ])-c)) for c in C]), x[1])))
                 .groupByKey()
                 .flatMap(point_count)
                 .reduceByKey(lambda x, y: x + y))
    triplets_list = triplets.collect()
    for c_id, N_vec in triplets_list:
        print(f"i = {c_id}, center = (", end = "")
        print("%.6f" % C[c_id][0], end = "")
        for i in range(1, len(C[c_id])):
            print(",%.6f" % C[c_id][i], end = "")
        print(f"), NA{c_id} = {N_vec[0]}, NB{c_id} = {N_vec[1]}")
```

The local space is expected to be asymptotically $M_L = \mathcal{O}(KN/L) = \mathcal{O}(K\sqrt{N})$, as $L = \sqrt{N}$ and where N is the number of points in U.

This result can be seen analyzing each operation on the Resilient Distributed Database:

1. The RDD triplets is constructed by first mapping points in U to their distance from the closest center. This mapping does not asymptotically impact the local space.

2. Resulting data is then randomly shuffled into L partitions. With high probability, the biggest of these is of size $\mathcal{O}(N/L) = \mathcal{O}(\sqrt{N})$.

3. For each partition we then count the number of points assigned to each center for the two groups A and B, reducing local space to $\mathcal{O}(K)$. As we assume $K \ll N$, this is asymptotically irrelevant.

4. Lastly we sum over all the partitions on a single node the number of points assigned to each of the K centers. As the number of partitions is $L = \sqrt{N}$ the local space needed is $\mathcal{O}(K\sqrt{N})$.

Finally the resulting RDD is collected locally to be displayed, but this is just a list of size $\mathcal{O}(K)$. So the maximum local size needed through the steps is $M_L = \mathcal{O}(K\sqrt{N})$.