**K-Means Algorithm Documentation**

**Hadar Pur - 308248533**

1. The master read and obtain all the points from the input file.
2. The master process does initial to the clusters according to the first k points.
3. The master process calculates the amount of points that each process (include him) will handle with, also, the master handle with the rest of the points (in case N%numproc!=0).
4. The master process broadcast to the other processes the number of clusters and number of points that the individual process will be handle.
5. The master process sends to each slave process that appropriate segment of points.
6. The all processes together (master and slaves) start the Algorithm.
7. The master process sends to each slave process the current time (OMP).
8. Each process calculates his own points coordinates (CUDA) according to the time from step 7.
9. Each process activate the K-Means function.

K-Means Algorithm:

* The master process sends the calculated clusters to the slave’s processes(OMP).
* Each process iterates on all his own points and finds the closest cluster to each of the points (OMP).
* Each process computes the sum of coordinates x, sum of coordinates y and sum of coordinates z of all the points who belong to the same cluster. Each cluster have his own sum X, sum Y and sum Z.(OMP)
* With the help of the MPI\_Reduce, all the sums of x, y and z are gathering in the master process.

Also, the total number of points who are belong to the same cluster are gathering together in the Master process (MPI\_SUM).

* The Master process, calculated and updated the clusters centers according to the sum x coordinates, sum y coordinates, sum z coordinates and the number of points who belong to specific cluster(OMP).
* Each process checks if his all of points belongs to the same cluster in step 2 of the K-Means algorithm.
* Each process sends the answer from the previous step to the master process.
* The master process gathers the answers from the step 6 with the help of the MPI\_Reduce and checks if the termination condition is fulfilled.
* If the termination condition is fulfilling or the master process has done limit iterations: the master process gathers all the points from all the slave processes, and send K-Means-Termination tag to the slaves and finally calculate and return the quality.
* If the termination condition is not fulfilled and the master process has not done limit iteration: return to step 1 in the K-Means algorithm.

1. The master obtains the current quality and checks if the quality is less than QM and check:
   * If the current quality is less than QM or the time is T/dt the master sends to all slave processes Final-Termination-tag and return the quality.
   * If the current quality is greater than QM, the master saves the less quality between the current quality and the previous quality.
2. The slaves’ processes are finished and finalize.
3. The master is writing to the output file the time, quality and the clusters centers to a output file and finalize.

**O((T/dt) \* Limit \* ( (N+K)/numOfProcs))**