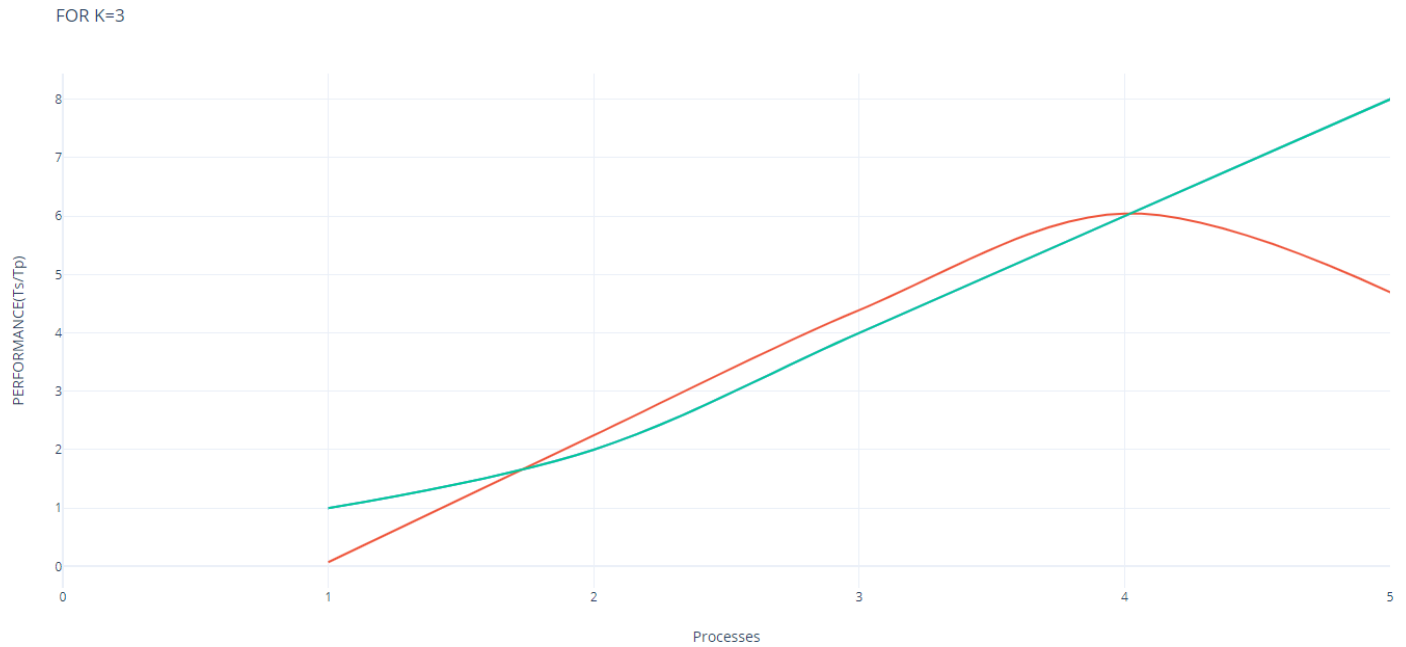


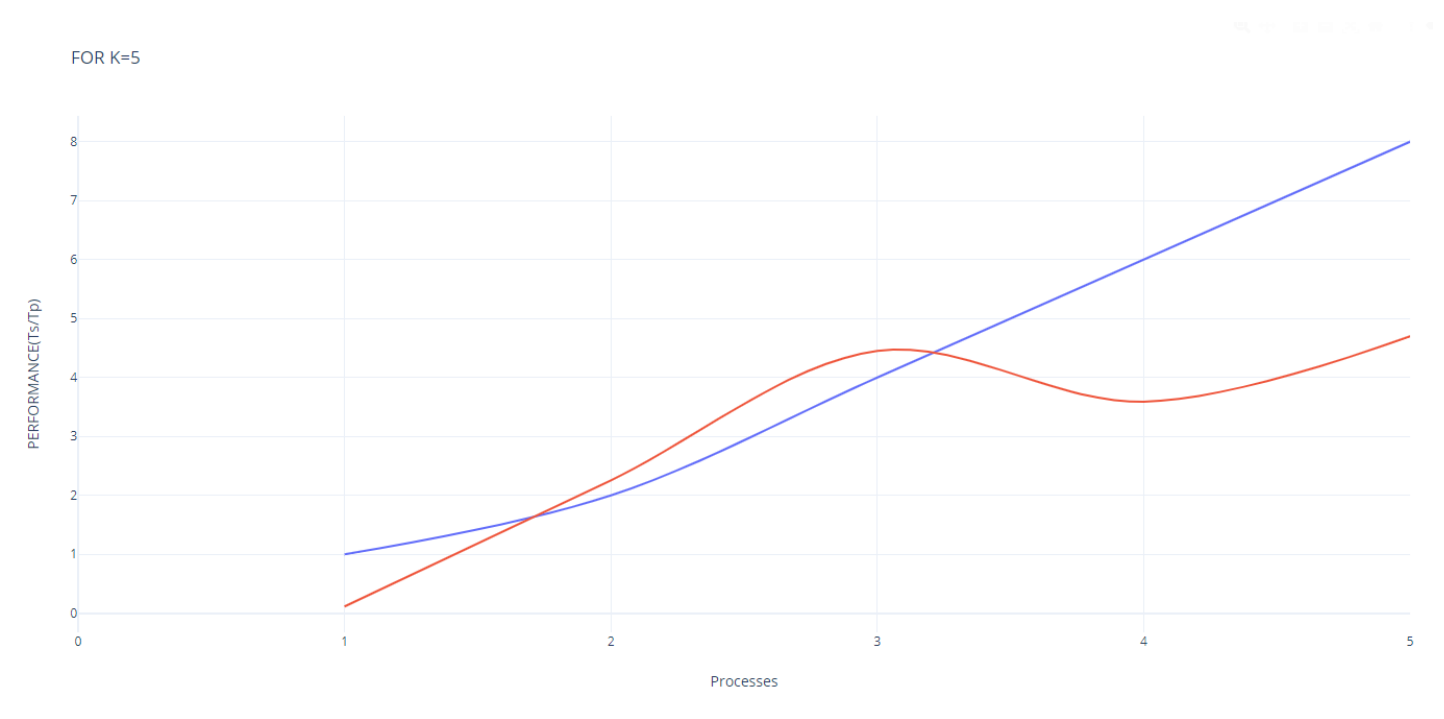
NAME-ISTIAQUE MANNAFEE SHAIKAT
MATRICULATION NUMBER-303527

For k=3



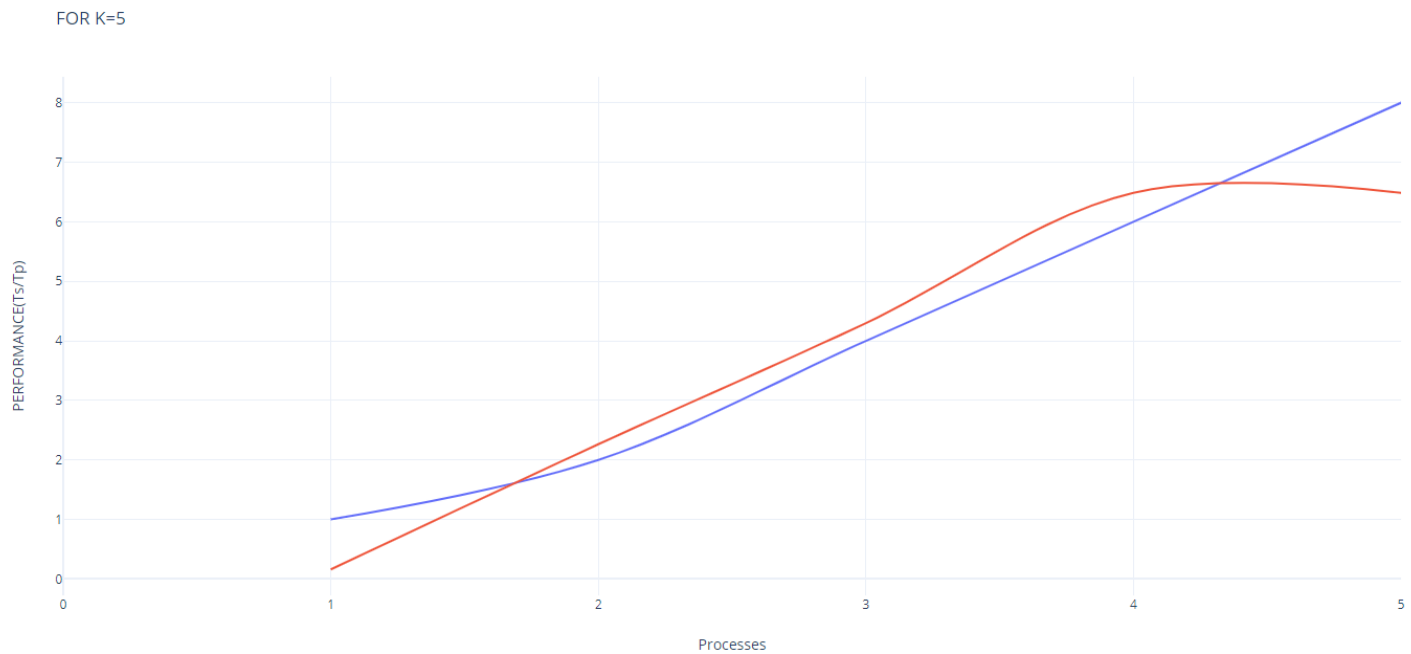
Performance Graph(serial		
Processes	execution/parallel	linear
1	0.0732938	1
2	2.245637779	2
4	4.389770278	4
6	6.044949134	6
8	4.69868194	8

For k=5



Processes	Performance	
	graph(serial/parallel)	linear
1	0.1158721	1
2	2.2557034	2
4	4.4508674	4
6	3.5902615	6
8	4.6985589	8

FOR K=7



Processes	Performance	
	graph(serial/parallel)	linear
1	0.1611017	1
2	2.2633382	2
4	4.2976842	4
6	6.4871949	6
8	6.4871949	8

Code description:

To perform distributed K mean clustering I have used collective communication which is scatter (distribute the chunk of data to all the workers) and broadcast (initial centroid, new centroid and global centroid)

First, worker 0 splits the data according to the size of the worker then this chunk of data is scattered to all the workers. Then the initial centroid is taken using the function `initial_centroid(data)` where full data is passed into the function. After that `k_mean(vA, vB)` is called where vA is the chunk of data and vB is the initial centroid.

In `k_mean` function we first find out the Euclidean distance of each data instances with first cluster to create a distance matrix then minimum distance's position is used to get the data instances and put it in the right cluster or K.

After assigning all the data instances in their right cluster I found the local mean of each chunk of data for specific cluster to calculate the new centroid which is the array `centers_new[]` and stored it in vC according to the number of worker. Later vC is used returned to 'new' variable to calculate the `global_centroid(new)`