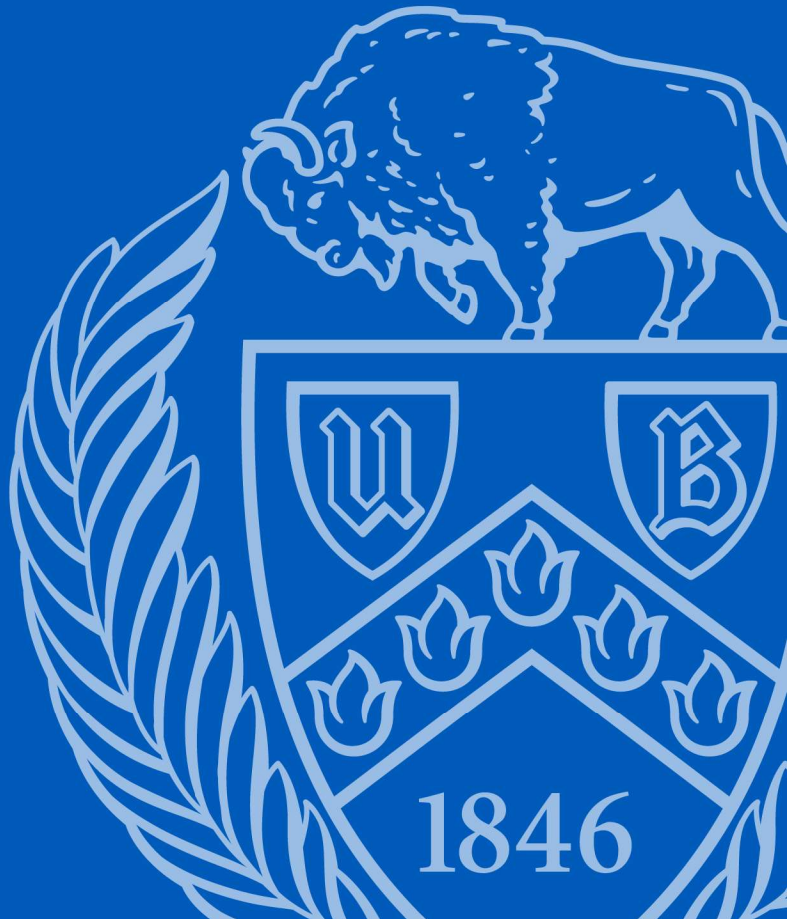
Image Segmentation using K-Means Algorithm

Aniket R Rane
CSE 633 Parallel Algorithms
Instructor – Dr. Russ Miller

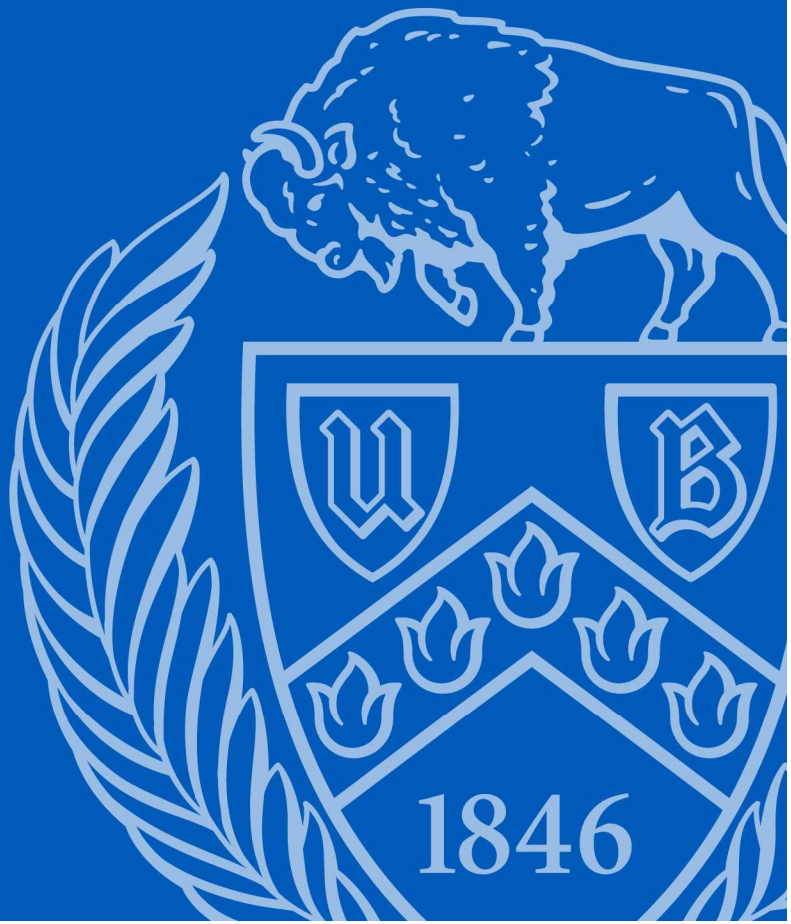


University at Buffalo The State University of New York

University at Buffalo The State University of New York

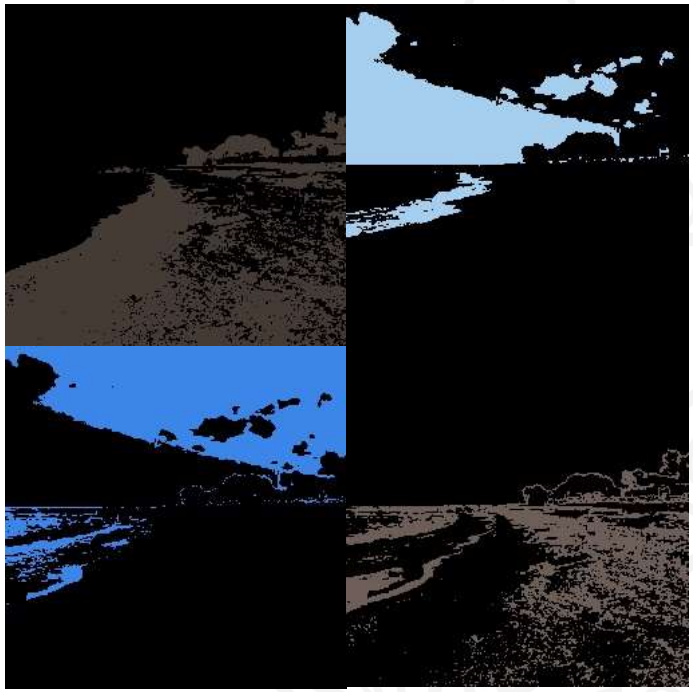
OUTLINE

- Proposed Project
- K Means
- Clustering
- Implementation
- Results
- Inferences
- Challenges
- References



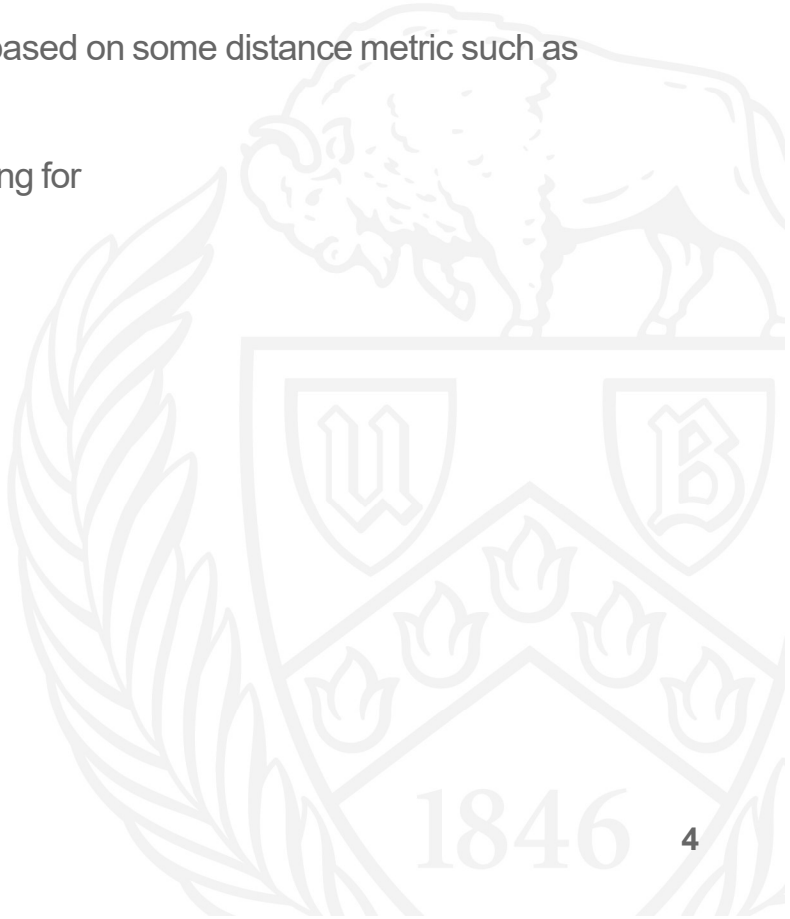
Proposed Project

- Image Segmentation using K-means : Break up the image into meaningful or perceptually similar images.



Clustering

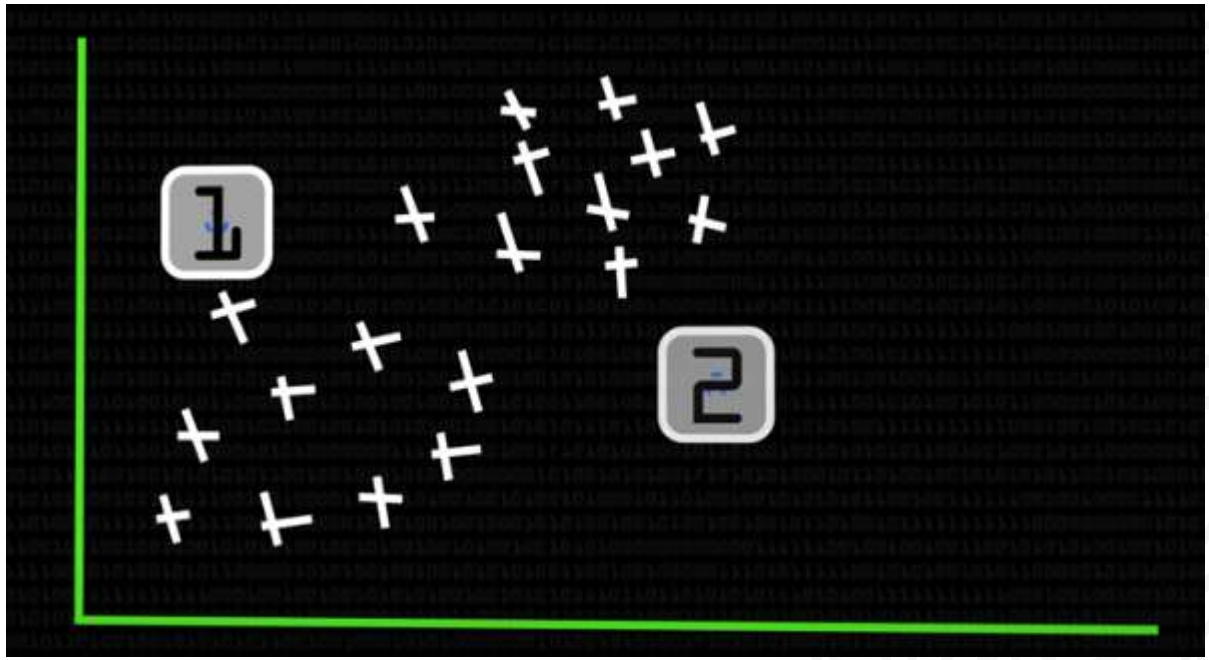
- Partitioning of data
- Similar elements placed in same cluster. Similarity is calculated based on some distance metric such as Euclidean distance
- Unsupervised Learning – Useful – Don't Know What you're looking for
- Requires data, but no labels
- Types
    - Partition Algorithms
    - Hierarchical Algorithms



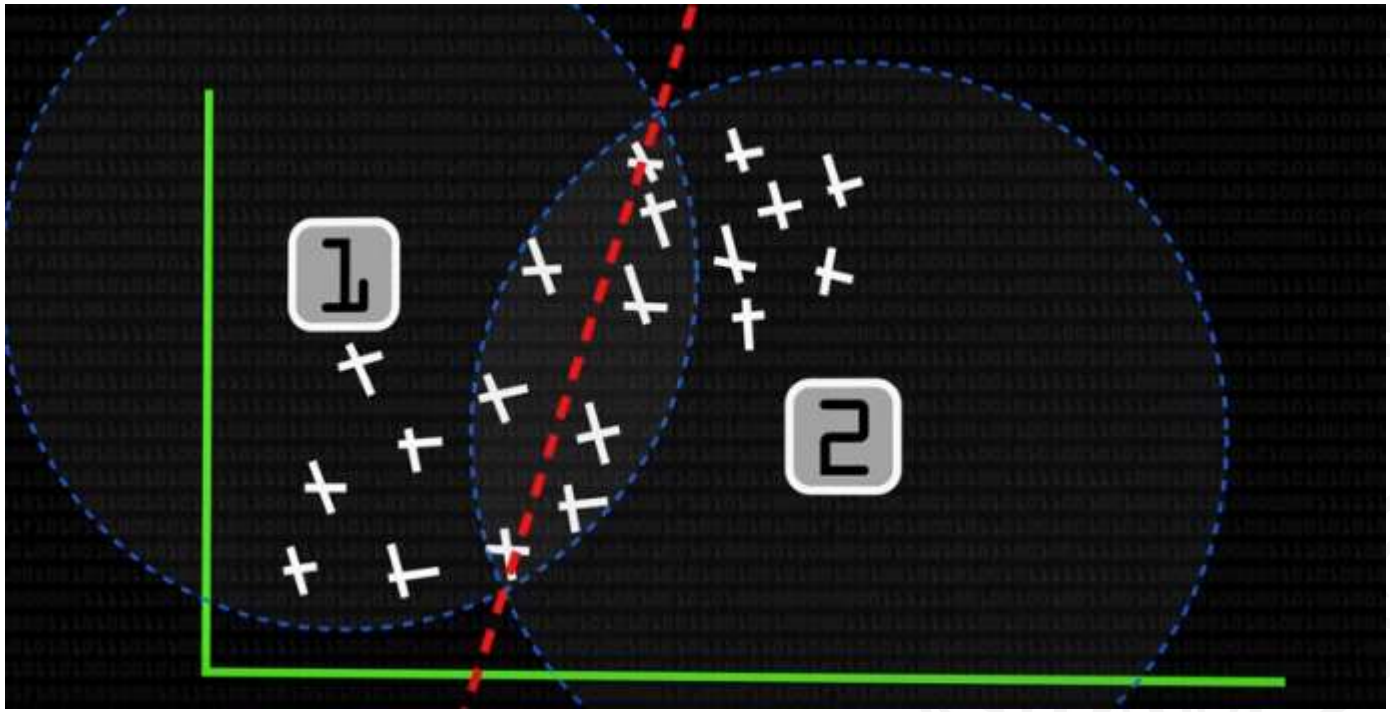
K-Means

1. Select k i.e. the number of clusters
2. Use a strategy to select k points to be cluster centers.
3. Put each point in the data set in the cluster which has its center closest to the point
4. Calculate new cluster centers by taking means of all points in a cluster
5. Repeat 3 and 4 until convergence

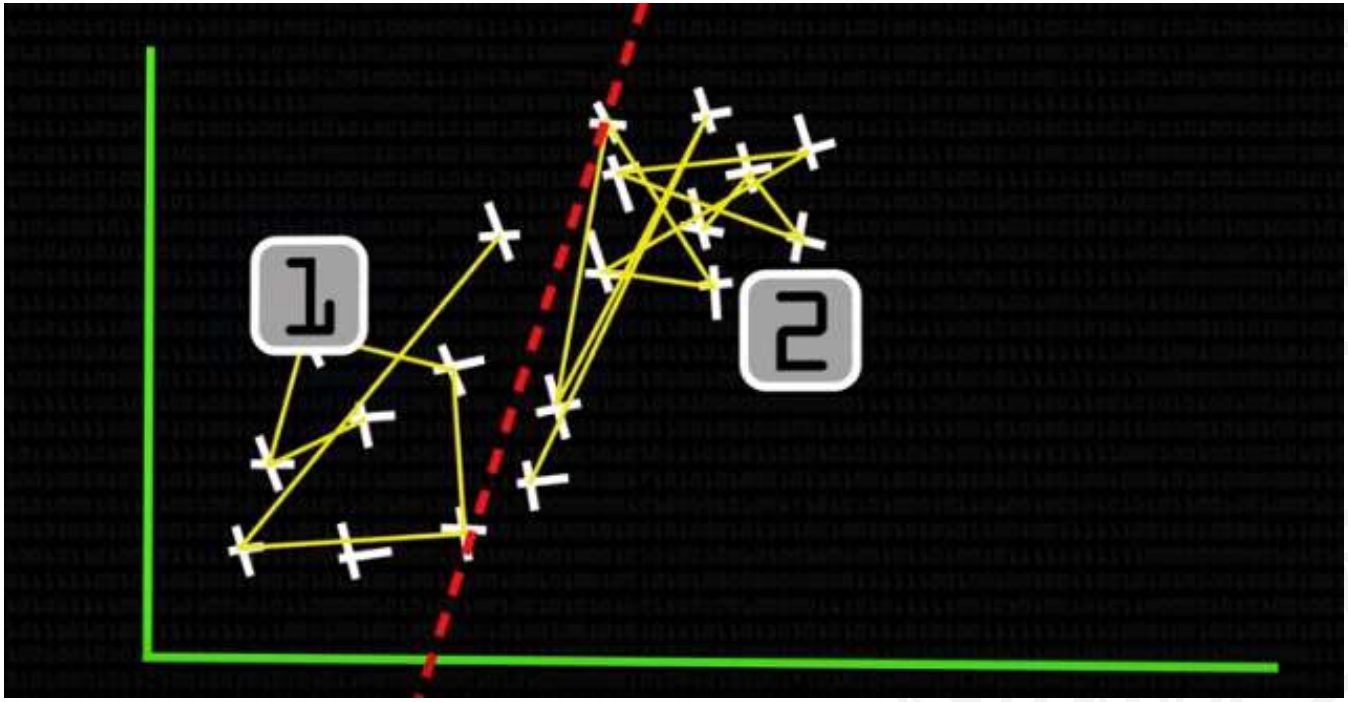
K = 2



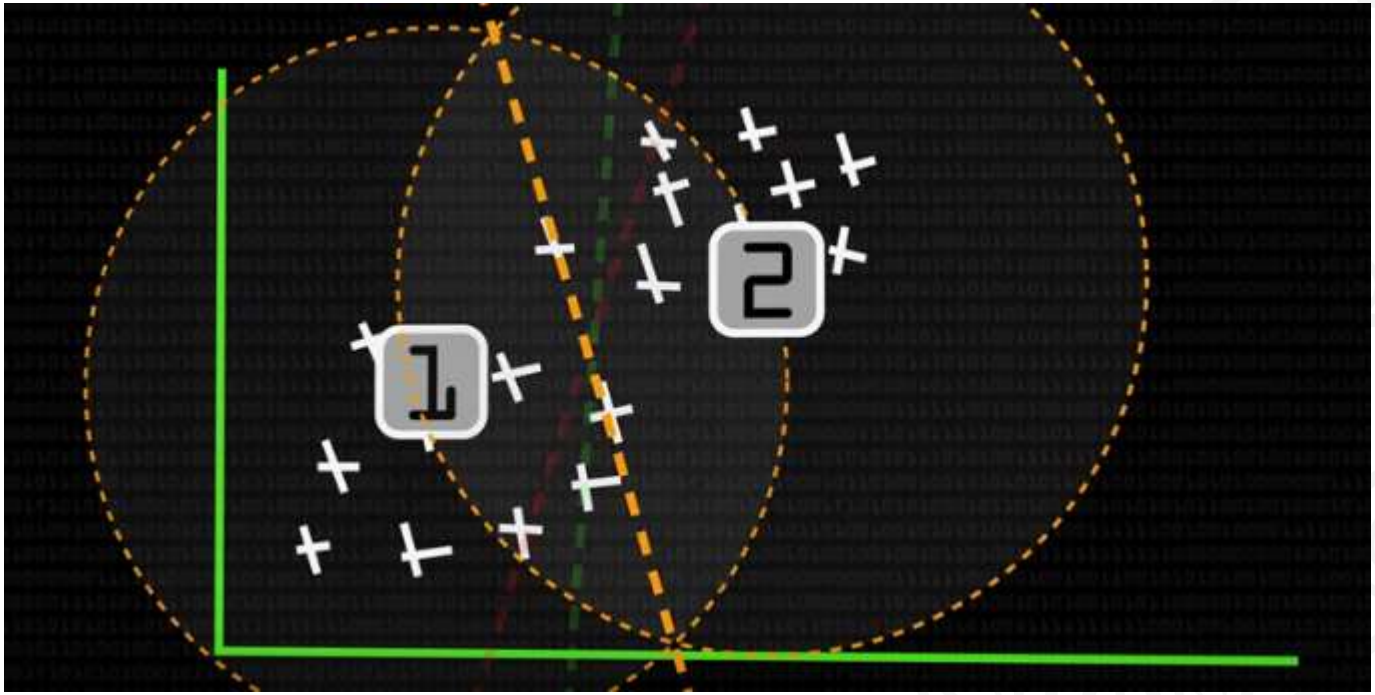
K = 2



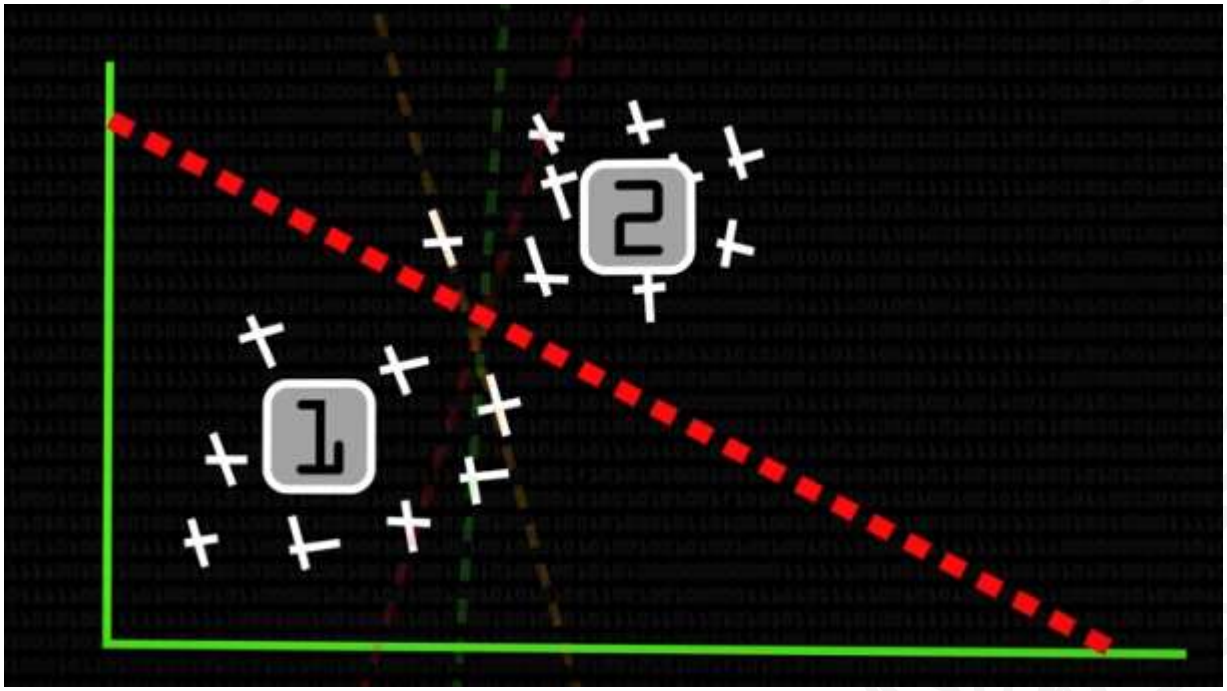
K = 2



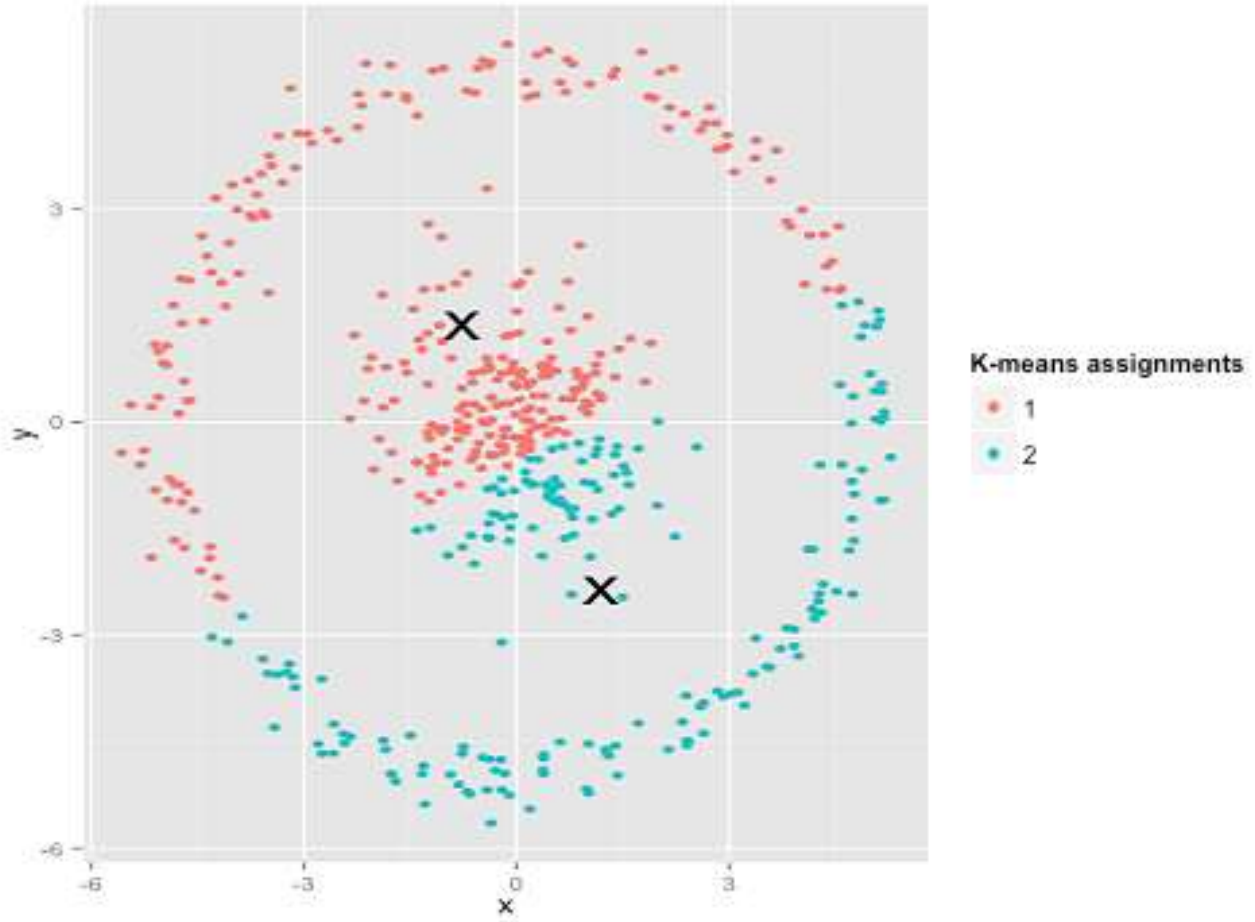
K = 2



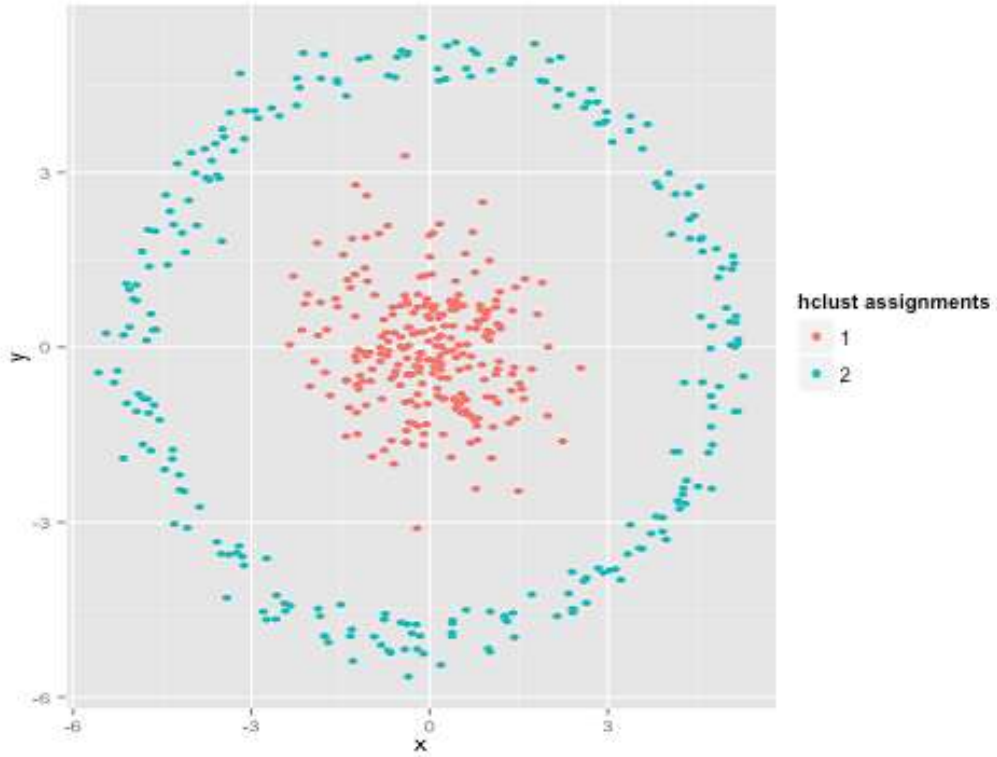
K = 2



Is K-Mean Ideal?



K-Means



Single Linkage, Hierarchical Clustering may work here!

Parallel Implementation – Image to Dataset

- Read the image using OpenCV for Python.
- Append the R, G, and B values of the pixels to a string one by one.
- Saving the string to a .txt file.

Parallel Model

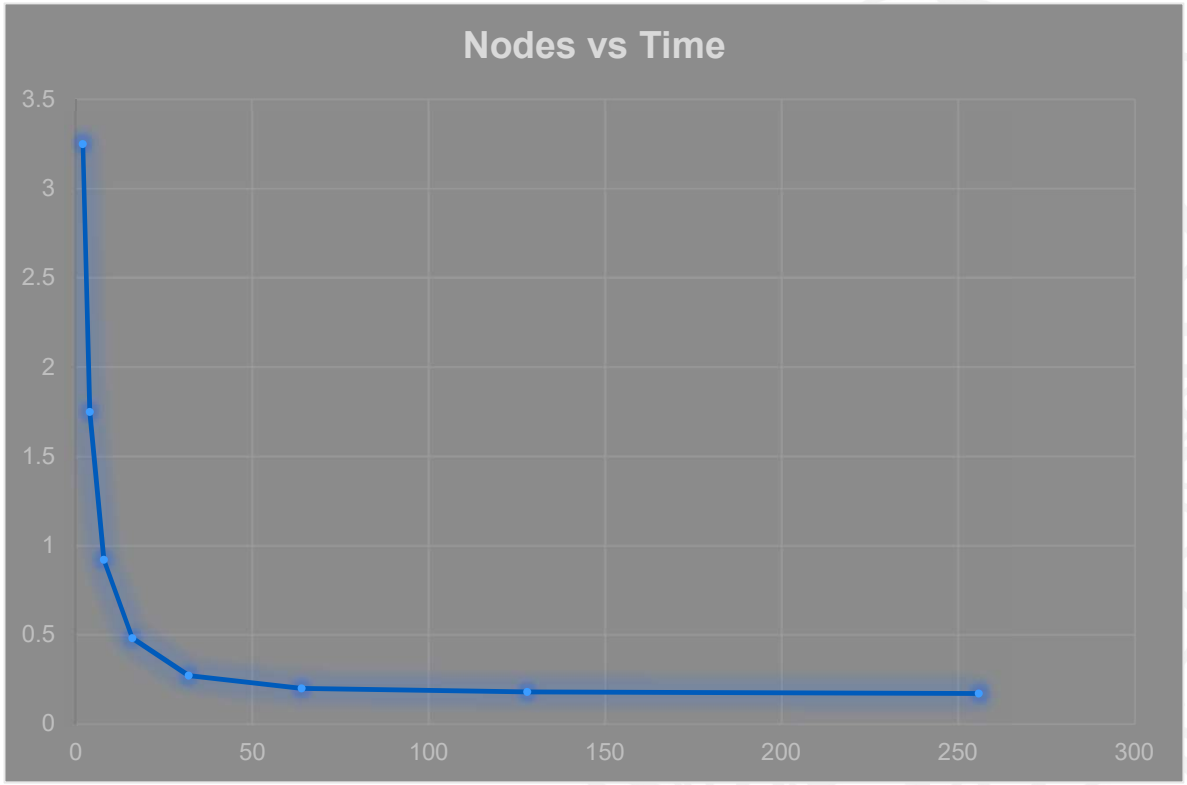
- Consider N data points and P processors.
- Assign N/P data points to each processor using .txt files.
- Node 0 randomly chooses k points as cluster centroids and broadcasts them.
- Each processor for each of its points, finds the cluster to which the point belongs.
- Recalculate local sums for each cluster in each processor.
- Send all local sums for each processor to processor 0 to find the global cluster centroids.
- Repeat the clustering for number of iterations.
- Save the cluster means of the final iteration.

Parallel Implementation – Independent Images of Clusters

- Read the file with final cluster means.
- Read the image.
- For each pixel, determine the cluster it belongs to and form a different image for every cluster with pixel values equal to the respective cluster means.
- Save the resulting images.

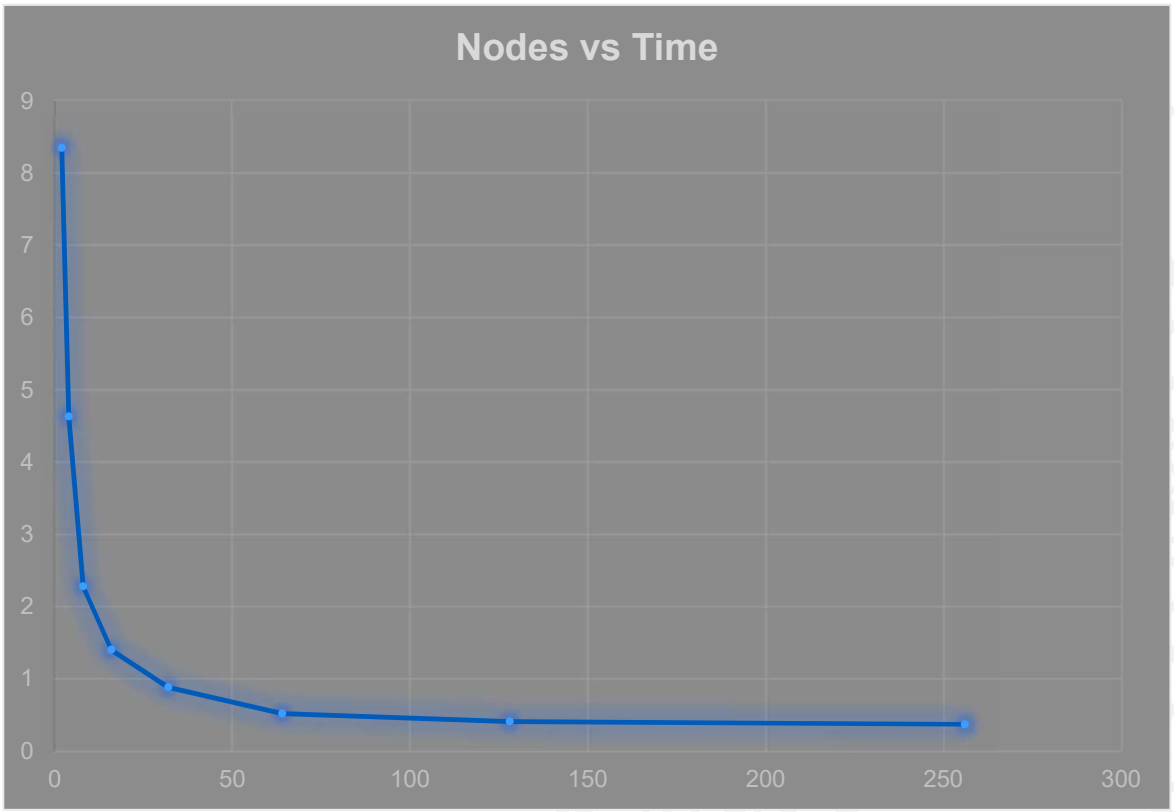
Results
3 Cluster 20 Iterations

Number of Processors	Time in seconds
2	3.25
4	1.75
8	0.92
16	0.48
32	0.27
64	0.2
128	0.18
256	0.17



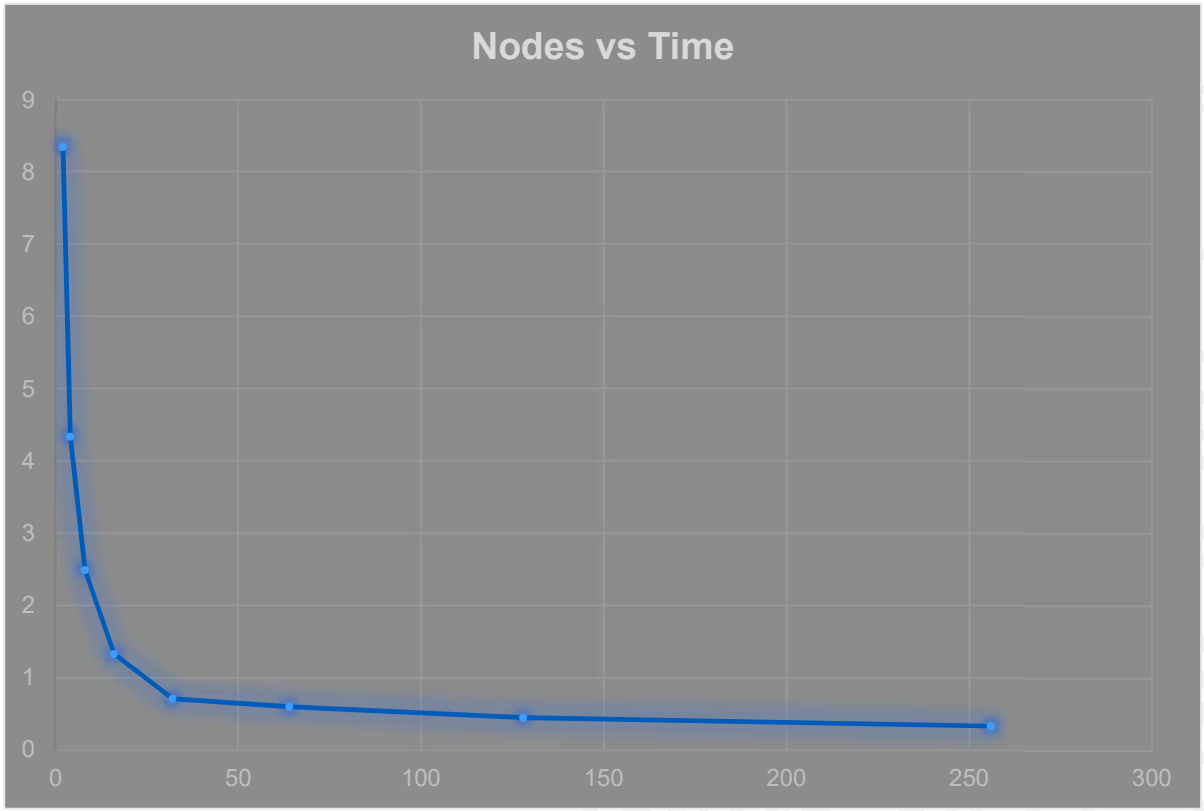
3 Clusters 40 Iterations

Number of Processors	Time in seconds
2	8.35
4	4.63
8	2.28
16	1.5
32	0.88
64	0.52
128	0.41
256	0.37



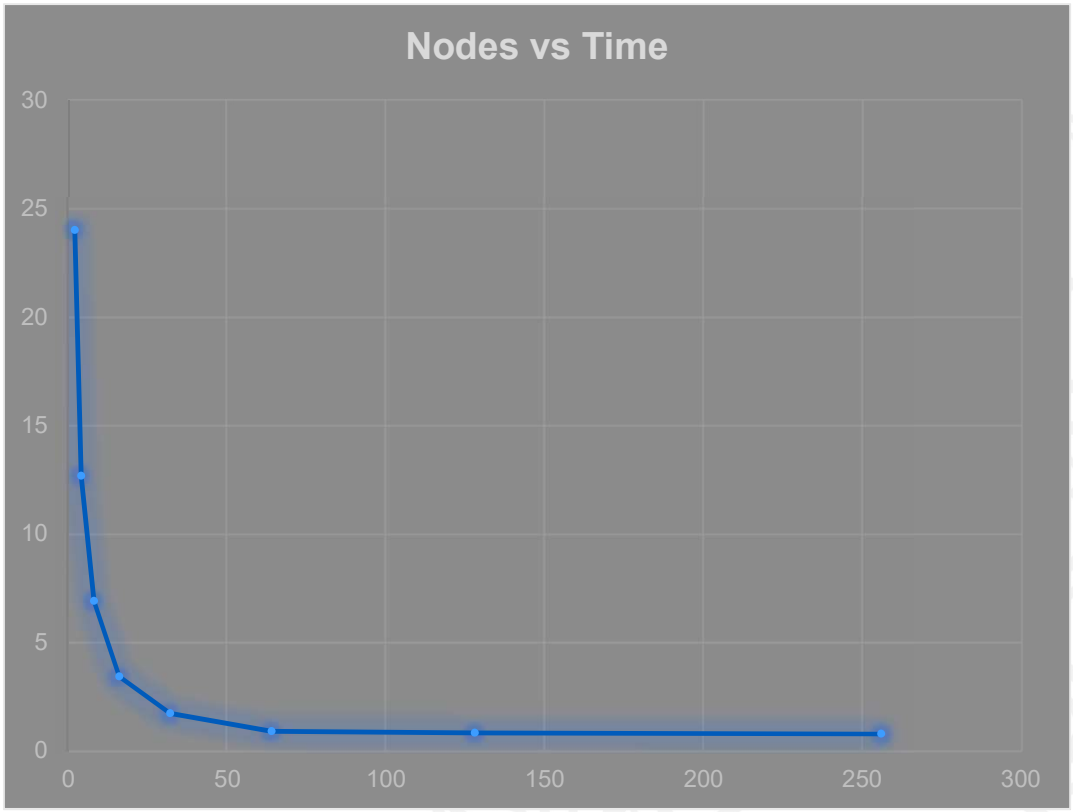
5 Clusters 20 Iterations

Number of Processors	Time in seconds
2	8.35
4	4.34
8	2.49
16	1.33
32	0.71
64	0.60
128	0.45
256	0.33

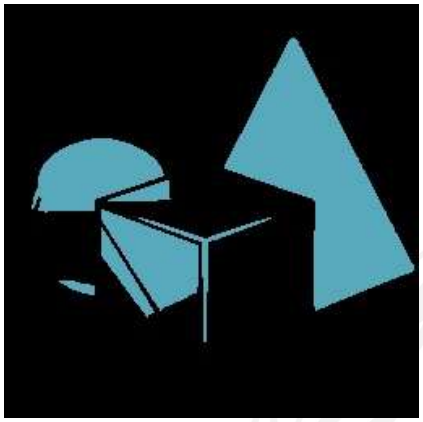


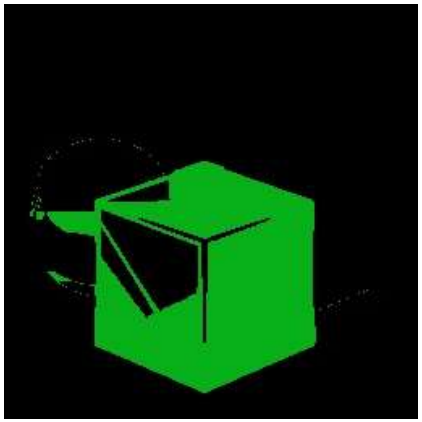
5 Clusters 40 Iterations

Number of Processors	Time in seconds
2	24.01
4	12.67
8	6.90
16	3.41
32	1.71
64	0.88
128	0.81
256	0.76

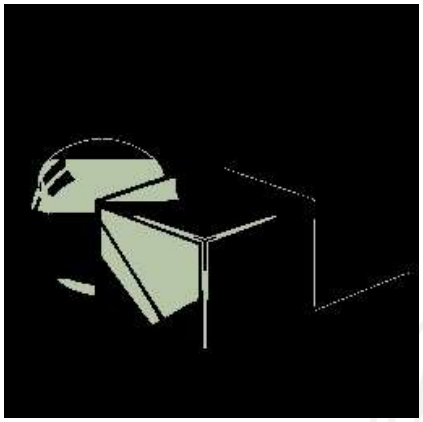


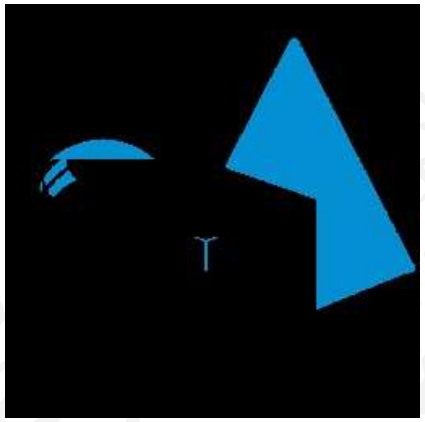
Independent 3 Clusters

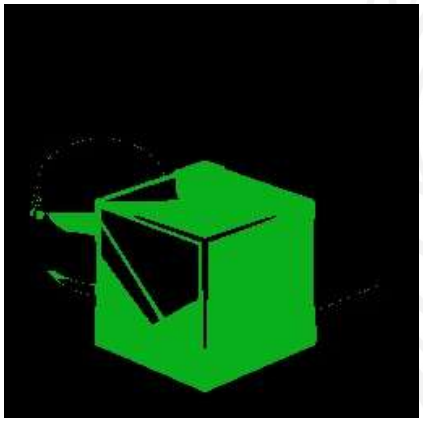




Independent 4 Clusters

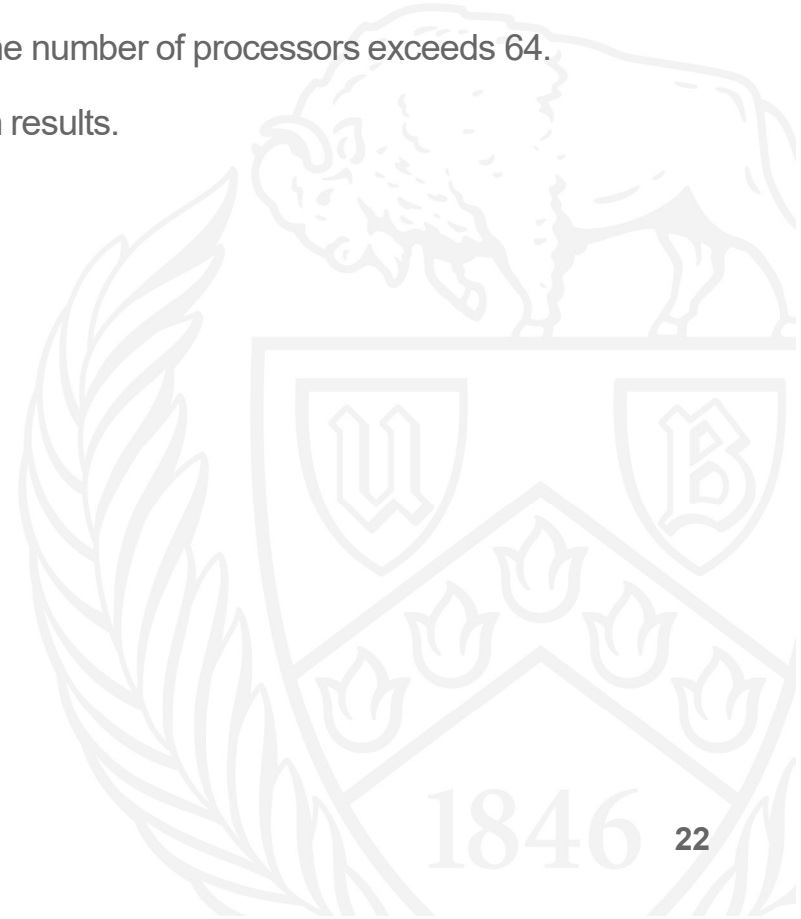






Inferences

- Significant speedup observed only up to 32 processors.
- Cost of communication affects the speedup significantly when the number of processors exceeds 64.
- Number of clusters has a big impact on the image segmentation results.
- Convergence is better after 40 Iterations



Challenges

- Time required for running the code for 256 processors was very high.
- Serializing the parallel components.
- Images compatible with K Means
- No support for image processing related libraries made it difficult to access image data directly and the image data had to be modified and stored in a .txt format.

References

- Algorithms Sequential & Parallel: A Unified Approach
  (Dr. Russ Miller, Dr. Laurence Boxer)
- https://ubccr.freshdesk.com/support/solutions/articles/13000026245-tutorials-and-training-documents
  (Dr. Matthew Jones)
- https://mpi4py.readthedocs.io/en/stable/tutorial.html#point-to-point-communication
- http://people.csail.mit.edu/dsontag/courses/ml12/slides/lecture14.pdf
- A Paralel K means clustering using MPI
  (Jing Zhang, Gongqing Wu, Xuegang Hu, Shiying Li, Shuilong Hao)
- Stackoverflow for general MPI questions