K-means is a popular unsupervised learning algorithm used to partition data points into clusters. With today’s abundance of data, datasets can easily reach sizes where local processing using the standard algorithm becomes an impossibility. In this paper, we present a cloud-based processing approach to implement K-means in order to handle datasets whose size prevents processing on a single machine.

**Abstract**

The algorithm can be described as follows:
1. Distribute $N$ instances evenly across all $P$ processors.
2. Initialize centroids using K-means ++ and send copy of centroids to all processors.
3. Repeat until convergence
   a) Perform standard K-means clustering on each processor
   b) Master processor receives all local centroid contributions from all $P$ processors, including itself and computes new centroids.
   c) Master processor broadcasts new centroids.

The K-means ++ initialization method provides a provable performance improvement against uniform centroid selection [2]. We incorporated this method into our design in order to improve the convergence rate.

In order to decrease execution time, we used cluster change flags. For each iteration, a cluster’s change flag is initialized to 0 and set to 1 if a data point leaves or joins the cluster. Before updating a centroid $c$, we bitwise-OR all $P$ change flags corresponding to the cluster defined by $c$, one from each of the $P$ processors. If the result is non-zero, we update it. If the result is 0, we can avoid recomputing the centroid. Important features of this implementation include:

- Data parallelism through MPI
- Task parallelism through OpenMP
- Prevention of unnecessary recomputing of unchanged centroids
- Centroid initialization using K-means ++

**References**


**Acknowledgements**

This research was sponsored by the National Science Foundation under grant CAHSI NSF CNS#1042341.