COL 380 **Jan 27, 2016**

Lab 1

Instructor: Subodh Sharma **Due: Feb 7, 23:55 hrs**

1 k-Means Clustering

K-means clustering is an important algorithm which finds its use in many areas of engineering. The goal of the algorithm is to find k partitions (or groups) in an unlabeled set of data points such that points within a partition have high similarity while the points belonging to different partitions have low similarity.

More formally, for a set of observations $(x_1, \cdots, x_n)$, where each observation is a d-dimensional real vector, k-means clustering aims to group the set of observations in to k sets $\{S_1, \cdots, S_k\}$ such that:

$$arg\ min \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{x,y \in S_i} \|(x-y)\|^2$$

.

1.1 Sequential Algorithm

A iterative sequential algorithm is provided for your reference:

1. For a given k, choose k centroids, $m_1, \cdots, m_k$. One can resort to *Forgy method* or *Random partition* method for this initialization.
2. After computing k centroids, assign points to each centroid to form a cluster. Common way to assign a point to a centroid is to find the nearest centroid. A point cannot, at any point, belong to two different clusters. At the end of this stage, one should have the clusters $\{S_1, \cdots, S_k\}$.
3. Recompute the mean (or the new centroid) m_j for each cluster S_j.
4. Step (3) will require re-partitioning of points since potentially new centroids were computed. Repeat steps (2) and (3) until convergence. Convergence conditions can be (i) maximum number of iterations, or (ii) if the total number of cluster membership changes in an iteration is below a threshold value.

1.2 Problem Statement

- Your task is to implement a parallel K-means clustering algorithm where the data points have a dimensionality of three and are of `int` type.
- Submit three versions of the code – one sequential, one with Pthreads, and one using OpenMP.
- Perform and plot speed-up and efficiency analysis of parallel implementations by changing the input size (i.e., number of threads and size of the dataset) of the problem.
- Submit a report clearly explaining your design decisions, parallelization strategy, loadbalancing strategy (if any). Make sure that explanations are bulleted and are not in the form of long paragraphs. Keep the lab report within 4 pages.

The input file format and the main file to calculate the time taken by your implementation will be provided at the earliest by the TAs. Make sure that the report and the code is not plagiarized. We have a repository of online codes. If you are found copying code from online sources, you will be penalized with a grade-drop + zero in the lab.

It is important that you start the assignment as early as possible. Waiting until the last few days to start the assignment, will make it extremely difficult for you to get help to complete the assignment.

1.3 Marking

1. Correct execution of the program	**50 Marks**
2. Performance of the program	**20 Marks**
3. Report	**30 Marks**