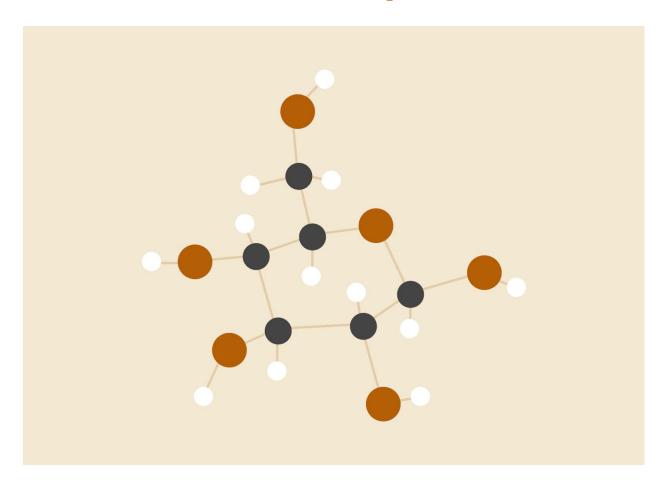
Parallel K-Means using Hadoop



Names:

Zeyad Ahmed Elbanna (26)

Ahmed Nasser abdelkareem Mohamed (9)

Abdelrahman Ahmed Mohamed Abdelfattah Omran (36)

The unparalleled K-Means pseudo-code.

K-Means Clustering

- 1. Choose the number of clusters(K) and obtain the data points
- 2. Place the centroids c_1, c_2, c_k randomly
- 3. Repeat steps 4 and 5 until convergence or until the end of a fixed number of iterations
- 4. for each data point x_i:
 - find the nearest centroid(c_1, c_2 .. c_k)
 - assign the point to that cluster
- 5. for each cluster j = 1..k
 - new centroid = mean of all points assigned to that cluster
- 6. End

MapReduce K-Means algorithm [1]

K-means using Map Reduce

Do Mar

Input is a data point and k centers are broadcasted Finds the closest center among k centers for the input point

Reduce

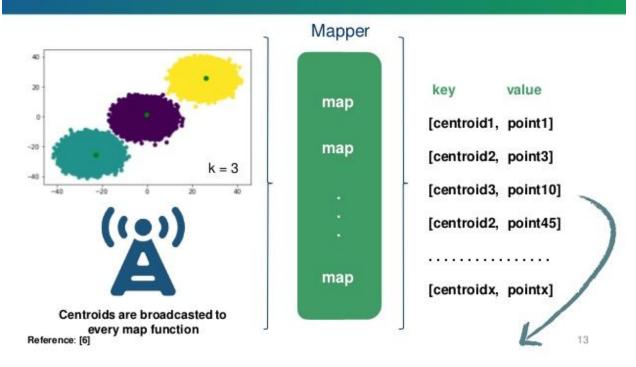
Input is one of k centers and all data points having this center as their closest center. Calculates the new center using data points

Until all of new centers are not changed

Reference: [6]

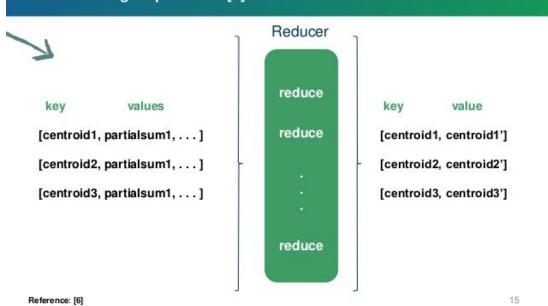
1) map function

K-means using Map Reduce [1]



2) Reduce-function

K-means using Map Reduce [3]



- The challenges you faced to implement it and how you solved it.
 - The namenode needed to be formatted (fixed)
 - Update centroids in array in runtime.
- The evaluation results (using a 1 node cluster is enough)
 - Parallel K-Means

Clusters				
0	5.006	3.418	1.464	0.24399
1	5.88	2.741	4.388	1.43
2	6.85	3.07	5.71	2.05

- In time: 14.272s with # of Iterations: 13

[1] Reference to slides which I implemented the code based on it

[slides]