TP 6 : Parallel implementation of Kmeans algorithm using mapreduce

Exercice 1 :

K-Means est un algorithme de clustering qui partitionne un ensemble de points de données en k clusters (Figure 1). L'algorithme de clustering k-means est couramment utilisé sur de grands ensembles de données et, en raison des caractéristiques de l'algorithme, est un bon candidat pour la parallélisation. Le but de cet exercice est d'implémenter k-means en utilisant Hadoop MapReduce.

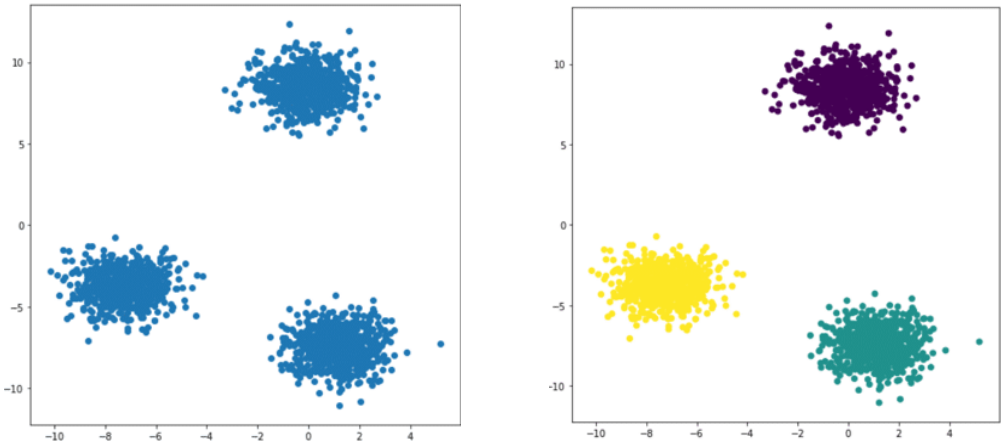

Figure 1: exemple de clustring avec k-means

L'algorithme est composé des étapes suivantes :

a) le processus de l'algorithme commence par sélectionner k centroïdes aléatoires;

b) les points sont affectés aux centroïdes les plus proches, en utilisant une mesure de distance sélectionnée;

c) la moyenne des points de chaque cluster est calculée et considérée comme nouveau centroïde;

d) les étapes 2 et 3 sont répétées jusqu'à ce que les centroïdes ne bougent plus ou le nombre maximum d'itérations a été atteint. En conséquence, les points sont finalement classés en k clusters.

Vous créez deux implémentations de l'algorithme :

1. la première est de faire le clustreing des points, chaque point est caractérisé par x et y. Vous créez un fichier csv contenant les points, chaque point est stocké dans une ligne, le format de fichier est le suivant :

	A	B
1	2	3
2	20	15
3	6	9
4		

Vous créez également un fichier centers.txt contenant les valeurs de trois centroïdes initiales, le fichier centers.txt est chargé dans le cache distribué de HDFS.

- La méthode map reçoit en paramètre comme valeur la ligne contenant les valeurs de x et de y du point, puis trouve le centroïde le plus proche au point en calculant la distance entre le point et tous les trois centroïdes, pour calculer la distance on utilise la formule suivante :

$$d = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

Dans l'opération reduce vous recevez en paramètre comme clé la valeur de centroïde de cluster et comme valeur une liste des point appartenant à ce cluster. Puis vous calculer la nouvelle valeur de centroïde qui représente la moyenne des point du cluster. Ensuite il faut refaire les même étapes jusqu'à la convergence de l'algorithme . la convergence et considérée quand les valeurs des centroïdes ne change plus.

- La deuxième implémentation de kmeans et de faire le clustering d'une image IRM cérébrale (Figure 2). L'image est en niveau de grille (grayscale image) ou la valeur de chaque pixel est entre 0 et 255 (0 représente la couleur noire et 255 représente la couleur blanche), cette image montre trois parties du cerveau à savoir la matière blanche, la matière grise et le liquide céphalorachidien, l'objectif et de savoir les pixels de chaque partie.

  Pour cela vous chargez l'image qui doit être stockée dans HDFS, puis classer l'image en trois clusters, vous créer également un fichier centers.txt qui contient trois valeurs des centroïdes initiales.

  La sortie du programme ça va être un fichier contenant le valeur de centroïdes des clusters et la liste des pixels apparentant à chaque clusters.

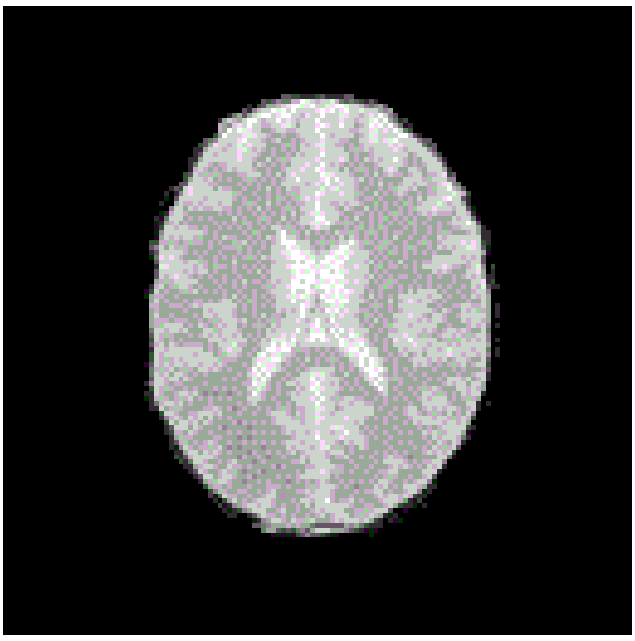

Figure 2: Image IRM cérébrale