**Série de TD n°1 :**

**MapReduce**

**Exercice n°1 :** Nous avons vu en cours l’algorithme qui définit en pseudo-code les classes Mapper et Reducer pour le calcul du nombre d’occurrences de mots dans une collection de documents. On considère la collection de paires clef/valeur suivante :

**Doc1** « *appreciate the fold* »

**Doc2** « *cata equals fold* »

**Doc3** « *fold the fold* »

1. En supposant que les documents **Doc1** et **Doc2** se situent sur un premier nœud et **Doc3** sur un autre, donner les résultats intermédiaires obtenus par chaque Mapper.

2. En supposant que l’espace des mots soit divisé en deux, les mots inférieurs à « *f* » sur un nœud et ceux après « *f* » sur un autre, donner les résultats intermédiaires obtenus des Reducer qui regroupent les clefs puis les résultats finaux des Reducer. Discuter la pertinence de cette règle de répartition des clefs dans ce cas d’application.

**Exercice n° 2 :** Soit une entreprise de grande distribution qui souhaite faire des statistiques sur les ventes des dix dernières années en ce début d’année. Elle possède une base de données stockée sur un système HDFS ; ne sera considérée que la partie de la base qui contient les données des dix années précédentes. Ces données sont stockées dans des fichiers textes. Chaque ligne d'un fichier correspond à la vente d'un produit et on peut y trouver des informations comme :

- la date (et l'heure) de vente

- le nom du magasin où le produit a été vendu

- le prix de vente

- la dénomination du produit

- la catégorie du produit (ex : fruits et légumes, électroménager, jouet, ....)

Exemple : 17\_12\_2017 hello 700 carteSD electronique

Écrivez le pseudo-code du programme map-reduce permettant de calculer le chiffre d'affaire de l'entreprise pour chacune des années précédentes, c'est à dire la somme total des ventes des 12 mois de l’année considérée.

**Exercice n°3 :** Nous considérons des données issues de diverses stations d’observation au tour du monde qui collectent des données météorologiques à chaque heure ; ceci engendre des quantités de données immenses. Ces données sont présentées sous forme de ligne de texte, où chaque ligne correspond à un enregistrement. Pour des raisons de simplification, nous n’allons présenter que quelques données. Voici dans ce qui suit l’exemple d’un enregistrement où nous ne présentons que quelques informations. Notons que tous les champs ont des tailles fixes.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id Station | Date | Heure | Latitude | Longitude | … | Température(Celsius \* 10) |  |
|  |  |  |  |  |  |  |  |
| 332130 | 19500101 | 0300 | +51317 | +028783 | … | -0128 | … |
| 0 |  |  |  |  |  | 88 |  |

Il vous est demandé d’écrire en Pseudo code les fonctions « Map » et « Reduce » qui permettent de retourner la température maximale pour chaque année.

**Exercice n°4 :** Soit un ensemble de fichiers contenant des mots. Ecrire les fonctions Map et Reduce (pseudo code) qui établissent la liste des mots, et pour chaque mot une liste des fichiers où il apparait.

Exemple :

|  |  |  |
| --- | --- | --- |
| **F1 : Ma ville Annaba****F2 : Annaba ville Plage Montagne** |  | **Annaba : F1 F2****Ma : F1****Montagne : F2****Plage : F2****Ville : F1 F2** |

**Exercice n°5 :** On considère des mesures regroupés dans des Records, stockés dans un fichier Hadoop (dans le système de fichiers HDFS), et représentant les sorties de différents capteurs : On peut ainsi analyser de multiples mesures concernant l’ensemble des moteurs. Afin d’obtenir un
système plus fiable, plusieurs capteurs sont installés sur chaque moteur, qui mesurent la même
grandeur, mais qui n’ont pas tous la même fiabilité.

moteurID, IdCapt1 : ValCapt1, IdCapt2 : ValCapt2, …, IdCaptn : ValCaptn

1. Ecrire, en pseudo code, les fonctions Map et Reduce qui permettent de calculer pour chaque moteur la valeur moyenne enregistrée par les capteurs.

En fait, toutes les mesures ne sont pas fiables. Il arrive qu’un capteur soit perturbé et produise une
valeur aberrante. Nous considérerons qu’une valeur de mesure inférieure à ‐100 ou supérieure à +100 est une valeur aberrante.

1. Concevez une solution, en pseudo code, Map‐Reduce qui pour un moteur donne la valeur moyenne enregistrée par ses capteurs non défaillants, ainsi que la liste des capteurs défaillants.

**Exercice n°6 :** Une entreprise souhaite réaliser des statistiques sur son personnel. Le fichier contient les informations relatives à chaque salarié (identifiant, age, sexe, adresse, salaire, nbrenfants...). Ecrire en pseudo-code les tâches Map et Reduce qui permettent de déterminer: Pour chaque âge, le salaire minimum, maximum et le nombre de personnes de cet âge.

**Exercice n°7 :** Ecrire en pseudo-code les fonctions Map et Reduce qui comptent le nombre de voyelles et celui des consonnes dans un texte en entrée.