MASTER 2 RECHERCHE INFORMATIQUE IAC

Module : Big Data & the Web

Rapport du projet Hadoop (Map/Reduce)

Réalisé par :

Abdelfettah Salim

Zaki Abdelwahed

2013 / 2014

En se basant sur un jeu de données RDF en format de triplets sujet, propriété, objet (**s p o.**), nous avons conçus les programmes Map/Reduce suivants :

1) Le nombre d'apparitions de chaque URI et chaque littéral

Un seul programme Map/Reduce qui est comme suit :

Fonction Map :

A partir du jeu de données retourne des paires de clés/valeurs : la clé est l'URI ou le Littéral et la valeur est le type défini sur un entier («0» pour sujet, «1» pour propriété et «2» pour objet).

Fonction Reduce :

A partir des paires obtenus précédemment et pour chaque URI/Littéral, calcul le nombre d'itération de la valeur en «0», «1» et «2» (correspondant respectivement au nombre de fois où le URI ou le Littérale apparait comme étant le sujet, propriété et objet) et les retourne sous cette forme :

URI/ Littéral (clé)	Nombre de fois qu'il apparait comme étant (valeur)		
	sujet	propriété	objet
URI1	5	0	1
…	…	…	…
Littéral1	2	0	0
…	…	…	…

2) Les 10 propriétés les plus fréquentes dans le jeu de données

Un seul programme Map/Reduce qui est comme suit :

Fonction Map :

A partir des résultats obtenus du programme Map/Reduce de la première question, fait une sélection sur les colonnes URL/Littéral et nombre de fois ou l'URI/Littérale apparait comme étant une propriété, et les retourne inversés en paires : clé (nombre de fois ou l'URI/Littérale apparait comme étant une propriété) et valeur (URL/Littéral).

Fonction Reduce :

Elle prend les paires obtenues de la fonction Map (le trie est fait sur la clé par ordre décroissant dans la phase de trie). Par la suite ne considérant que les 10 premières paires (*ou un peu plus si par exemple le 10ème et le 11ème ont la même valeur sur la clé*), elle retourne encore une fois inversées (pour les avoir dans le bon ordre en quelque sorte) ses paires comme ceci :

Propriété	Nombre d'apparition
P1	20

P2	19
…	…

3) Les 10 classes apparaissant le plus fréquemment dans le jeu de données

Pour cette question cela a nécessité deux programmes Map/Reduce qui sont comme suit :

1ère Fonction Map :

A partir du jeu de données et plus précisément les valeurs des propriétés, cherche ce qui peut identifier une classe (inclusion de «type», «subClassOf», «domain» et «range») et selon les cas retourne des paires de clés/valeurs (la clé est l'URI/Littérale qu'il soit sujet ou objet et la valeur un entier) comme suit :

⟹ sujet/objet est une classe, retourne (URI/Littéral, -1)
⟹ sujet/objet n'est pas une classe, retourne (URI/Littéral, 1)
⟹ pour <rdfs:Class> ne retourne rien

1ère Fonction Reduce :

Prend les paires de clés obtenues du Map précédant et détecte si un sujet/objet est une classe en vérifiant si on le trouve au moins une fois avec la valeur -1 (ceux n'ayant que des valeurs 1 ne sont alors pas des classes et sont ignorés). La fonction retourne les paires comme suit : clé : classe (URI/Littérale), valeur : nombre d'apparitions (comme étant classe ou non).

Classe	Nombre d'apparitions
C1	20
C2	19
C3	22
…	…

2ème Fonction Map :

Prend les paires obtenues du 1er Map/Reduce et les retourne inversées (nombre d'apparitions, classe).

2ème Fonction Reduce :

Filtre les paires reçus du 2ème Map (trié par la clé par ordre décroissant) en ne laissant passer que les 10 premières paires. Les paires sont retournées en ré inversés (pour avoir le bon ordre) comme suit :

Classe	Nombre d'apparition
C3	22
C1	20

C2	19
…	…

4) Les 10 sujets ayant le plus grand nombre de propriétés distinctes spécifiées

Pour cette question cela a nécessité deux programmes Map/Reduce qui sont comme suit :

1ère Fonction Map :

A partir du jeu de données sélectionne les paires (Sujet, Propriété) et les retourne comme clé/valeur.

1ère Fonction Reduce :

Prend les paires obtenus du Map (sachant qu'elles ont été ordonnées par la valeur) et compte le nombre de propriétés distinctes (en ignorant la duplication des propriétés ; cela est dû d'une part au trie par valeur) de chaque clé et retourne les paires (sujet, nombre de propriétés).

Sujet	Nombre de propriétés
S1	15
S2	21
S3	10
…	…

Le deuxième Map/Réduce est similaire à celui du trisième programme et retourne les 10 premières paires comme suit :

Sujet	Nombre de propriétés
S2	21
S1	15
S3	10
…	…