UE INF2245
Examen final de TP sur Hadoop

Frédéric Raimbault

Avertissement : ceci est un examen. Tous les programmes comportants des similarités flagrantes verront, a minima, leurs auteurs sanctionnés par un 0 à l'UE (et pas seulement au TP).

Top10 des contributeurs de StackOverflow

L'objectif du TP est de refaire le dernier exercice du TP précédent, "déterminer quel est le top10 des utilisateurs qui ont eu le plus de réponses acceptées", en écrivant 3 programmes MapReduce consécutifs (les programmes MR seront lancés successivement depuis la ligne de commande). Le premier programme MapReduce prend en entrée le fichier `Posts.xml` (en pratique vous prendrez `Posts-100K.xml` ou `Posts-1M.xml` pour vos tests). Les programmes suivants prennent en entrée le résultat du programme précédent. Le dernier programme produit la liste ordonnée des *Id* des 10 premiers auteurs, avec, pour chacun, le nombre de réponses acceptées.

1. AcceptedJoin : parcours des posts et pour chaque question récupération de l'*Id* de l'auteur de la réponse acceptée[1]
2. AcceptedCnt : comptage du nombre de réponses acceptées de chaque auteur
3. AcceptedTopTen : classement et sélection du top10 des auteurs.

Rappel de quelques champs utiles de `Posts.xml` (cf `readme.txt` sur l'ENT) :

Id contient l'*Id* du *post*.

PostTypeId contient le type de post : *"1"* pour une Question, *"2"* pour une réponse.

AcceptedAnswerId contient l'*Id* du *post* contenant la réponse acceptée (si c'est une question).

OwnerUserId contient l'*Id* du rédacteur du *post*.

[1] C'est une opération de jointure entre les questions et les réponses acceptées.

Pour l'extraction d'une valeur d'attribut vous utiliserez la méthode XmlUtils.getAttributeValue()

Voici quelques détails sur les 3 programmes MapReduce enchainés (les *Driver* vous sont fournis sur l'ENT) :

1. AcceptedJoinDriver, AcceptedJoinMapper, AcceptedJoinReducer

   - en entrée : les posts (prendre `Posts-100K.xml` ou `Posts-1M.xml` pour vos tests)
   - Les mappers extraient des posts qui sont des questions l'*Id* du post de la réponse acceptée et extraient des posts qui sont des réponses l'*Id* du post et l'*Id* de l'auteur. Vous gèrerez les compteurs suivants dans le mapper (leur valeur devra apparaitre en commentaire du source Java du mapper) :

```
// recopier dans AcceptedJoinMapper
public static enum MapCounters {
/* valeurs obtenues sur Posts-100K.xml / sur Posts-1M.xml */
        QUESTIONS_NB,       /* valeur =  / */
        RESPONSES_NB,       /* valeur =  / */
        UNKNOW_POST_TYPE,   /* valeur =  / */
        INVALID_POST_TYPE,  /* valeur =  / */
        UNKNOW_ANSWER_ID,   /* valeur =  / */
        UNKNOW_ID,          /* valeur =  / */
        UNKNOW_OWNER_ID,    /* valeur =  / */
};
```

   - les reducers fusionnent les posts (version très simplifiée du Pattern *innerjoin*) sur leur Id et extraient l'Id de l'auteur.
   - en sortie : les couples $(userid, 1)$ pour chaque auteur d'un post contenant une réponse acceptée.

2. AcceptedCntDriver, AcceptedCntReducer :

   - en entrée : le résultat de AcceptedJoinDriver.
   - les mappers sont des fonctions identité et produisent donc des couples $(userid, 1)$.
   - les reducers comptabilisent le nombre de réponses acceptées par chaque auteur. Vous gèrerez les compteurs suivants dans le reducer (leur valeur devra apparaitre en commentaire du source Java du reducer) :

```
// recopier dans AcceptedCntReducer
public static enum ReduceCounters {
/* valeurs obtenues sur Posts-100K.xml / sur Posts-1M.xml */
        USER_CNT, /* valeur =  / */
        SUM_CNT,  /* valeur =  / */
};
```

- en sortie : les couples $(userid, count)$ pour chaque auteur d'un post d'une (au moins) réponse acceptée.

3. AcceptedTopTenDriver, AcceptedTopTenMapper, AcceptedTopTenReducer :
   - en entrée : le résultat de AcceptedCntDriver, c-à-d. des couples $(userid, count)$
   - en sortie 10 couples $(count, userid)$ ordonnés par nombre décroissant de nombre de posts. Vous donnerez cette liste en commentaire dans AcceptedTopTenReducer.
   - application directe du pattern TopTen (programmation par des TreeMap).