Plateformes et Langages de Programmation

PARTIE HADOOP MAP-REDUCE

CentraleSupélec

Zemeng SU & Lu YU & Hanqing ZHU

CENTRALESUPELEC | OMA

Les contributions respectives

Question 2.7
Code : Zemeng SU & Lu YU & Hanqing ZHU
Rapport : Lu YU

Question 2.8
Code : Zemeng SU & Lu YU & Hanqing ZHU
Rapport : Lu YU

Question 5.1
Code : Zemeng SU & Lu YU
Rapport : Zemeng SU

Question 5.2
Code : Hanqing ZHU
Rapport : Hanqing ZHU

Question 5.3
Code : Zemeng SU & Lu YU
Rapport : Lu YU

Ce projet PLP a pour objectif d'utiliser Hadoop sur une machine virtuelle Cloudera pour résoudre quelques problèmes de big data. On va surtout utiliser la méthode de MapReduce dans les questions.

Question 2.7 Displaying the content of a CSV file

L'objectif de cette question est d'afficher l'année et l'hauteur de chaque arbre depuis un fichier *arbres.csv.* Voici le résultat obtenu:

```
Year : 1935 & Height : 13.0
Year : 1854 & Height : 20.0
Year : 1862 & Height : 22.0
Year : 1906 & Height : 16.0
Year : 1784 & Height : 30.0
Year : 1860 & Height : 45.0
Year : 1840 & Height : 40.0
Year : 1933 & Height : 16.0
```

Question 2.8 Displaying the content of a compact file

L'objectif de cette question est d'écrire un programme qui peut afficher le USAF code, le nom, le pays et l'élévation de chaque station en utilisant le HDFS API avec le fichier *isd-history.txt*. Voici le résultat obtenu:

```
USAF: 999999 ,the name of the station:  MERCED 23 WSW                ,the country:  US ,the elevation:  +0023.8.
USAF: 999999 ,the name of the station:  BODEGA 6 WSW                 ,the country:  US ,the elevation:  +0019.2.
USAF: 999999 ,the name of the station:  NORTH MYRTLE BEACH AIRPORT    ,the country:  US ,the elevation:  +0010.1.
USAF: 999999 ,the name of the station:  NEW BERN CRAVEN CO REGL AP    ,the country:  US ,the elevation:  +0007.3.
USAF: 999999 ,the name of the station:  SALISBURY WICOMICO CO AP      ,the country:  US ,the elevation:  +0018.3.
USAF: 999999 ,the name of the station:  BALTIMORE BLT-WASHNG          ,the country:  US ,the elevation:  +0047.2.
USAF: 999999 ,the name of the station:  KINSTON STALLINGS AFB         ,the country:  US ,the elevation:  +0028.7.
USAF: 999999 ,the name of the station:  NEW RIVER MCAS                ,the country:  US ,the elevation:  +0007.6.
USAF: 999999 ,the name of the station:  HATTERAS BILLY MITCHELL AP    ,the country:  US ,the elevation:  +0003.4.
USAF: 999999 ,the name of the station:  ATLANTIC CITY INTL A          ,the country:  US ,the elevation:  +0020.4.
USAF: 999999 ,the name of the station:  NORFOLK FLEET WF NAS          ,the country:  US ,the elevation:  +0010.1.
USAF: 999999 ,the name of the station:  NEW YORK SHOALS AFS           ,the country:  US ,the elevation:  +0025.9.
USAF: 999999 ,the name of the station:  STERLING                      ,the country:  US ,the elevation:  +0085.0.
USAF: 999999 ,the name of the station:  FORT EUSTIS FELKER AAF        ,the country:  US ,the elevation:  +0003.7.
USAF: 999999 ,the name of the station:  CHARLOTTESVILLE FAA           ,the country:  US ,the elevation:  +0196.3.
USAF: 999999 ,the name of the station:  FORT BRAGG SIMMONS AAF        ,the country:  US ,the elevation:  +0073.8.
USAF: 999999 ,the name of the station:  WASHINGTON DULLES INTERNATION ,the country:  US ,the elevation:  +0098.5.
USAF: 999999 ,the name of the station:  WALLOPS ISLAND UAU            ,the country:  US ,the elevation:  +0014.6.
USAF: 999999 ,the name of the station:  BOGUE FIELD MCAF              ,the country:  US ,the elevation:  +0006.7.
USAF: 999999 ,the name of the station:  CHERRY POINT/MCAS ASOS 2      ,the country:  US ,the elevation:  +0009.1.
USAF: 999999 ,the name of the station:  ROME R B RUSSELL AP           ,the country:  US ,the elevation:  +0196.3.
USAF: 999999 ,the name of the station:  SMITHVILLE CAA AP             ,the country:  US ,the elevation:  +0328.9.
USAF: 999999 ,the name of the station:  SPARTANBURG                   ,the country:  US ,the elevation:  +0244.1.
USAF: 999999 ,the name of the station:  TALLAHASSEE DALE MABRY FIELD  ,the country:  US ,the elevation:  +0020.7.
USAF: 999999 ,the name of the station:  TUSCALOOSA MUNICIPAL AP       ,the country:  US ,the elevation:  +0056.7.
USAF: 999999 ,the name of the station:  WINSTON-SALEM REYNOLDS AP     ,the country:  US ,the elevation:  +0295.7.
USAF: 999999 ,the name of the station:  BOWLING GREEN WARREN CO AP    ,the country:  US ,the elevation:  +0163.7.
```

Question 5.1 TF-IDF

Le TF-IDF signifie *term frequency-inverse document frequency* est une méthode numérique statistique utilisé pour l'évaluation d'importance d'un mot dans un document par rapport à une collection de textes. Le TD-IDF est un produit de deux statistiques : TD et IDF. Nous allons définir ces deux termes avec les formules suivantes :

$$tf = \frac{WordCount}{Words\ per\ Document}$$

$$idf = \log(\frac{Total\ Documents}{Documents\ per\ Word})$$

La TF est une mesure de fréquence d'un mot dans un seul document. La IDF est une mesure concernant la quantité d'information portée par ce terme.

Notre programme contient six classes :
RoundOneMapper
RoundOneReducer
RoundOneCombiner
RoundOnePartionner
RoundTwoMapper
RoundTwoReducer
Tdidf

Dans la classe Tfidf se trouve la fonction *main* où nous avons défini les configurations et les jobs pour les deux tours de travaux (*job*).

Dans le premier tour de travaux (*RoundOne*), nous avons utilisé un *partitionner* et un *combiner.* Un *partitionner* a pour l'objectif d'améliorer l'équilibrage de charge les *reducers*. Un *combiner* est pour réduire le trafic de données afin d'améliorer le temps de traitement et la performance.

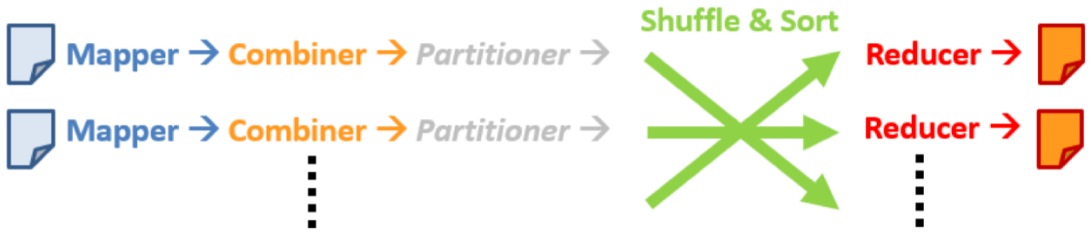


Figure 1 Digramme de procédure des jobs pour RoundOne

La figure présentée au-dessus est la procédure de traitement pour notre programme au tour n°1 (*RoundOne).*

Figure 2 Diagramme de procédure des jobs pour RoundTwo

Pour la deuxième tour de travaux, nous avons une structure simple qui continent seulement *Mapper* et *Reducer*.

Pour le RoundOne, nous avons précisé les entrées et les sorties de chaque composant de notre programme.

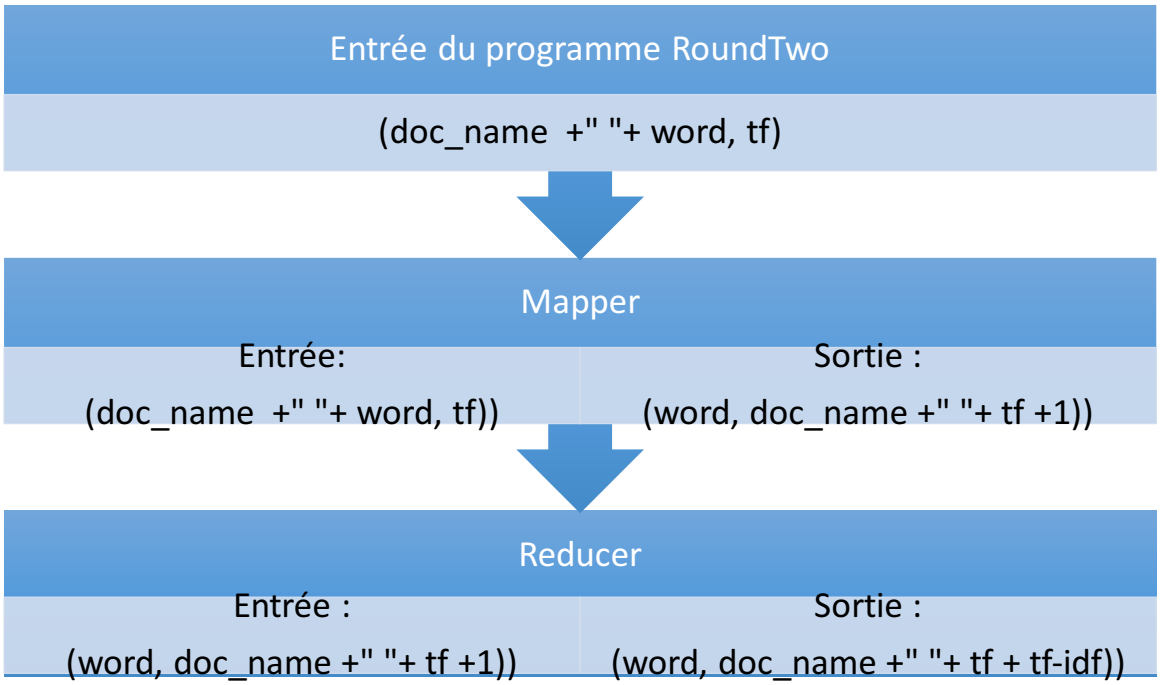
Entrée du programme RoundOne	
Doc 1	Doc 2

Mapper	
Entrée : (void, doc_name)	Sortie : (word +" "+ doc_name, 1)

Combiner	
Entrée : (word +" "+ doc_name, 1)	Sortie : (doc_name +" "+ word, tf)

Reducer	
Entrée : (doc_name +" "+ word, tf)	Sortie : (doc_name +" "+ word, tf)

Pour le RoundTwo, nous avons précisé les entrées et les sorties de chaque composant de notre programme.



La figure ci-contre est une petite partie de notre résultat final.

```
Also    callwild        3.1468313E-5 -9.472906112255005E-6
Also    defoe-robinson-103.txt  3.290908E-5 -9.906620177012755E-6
Also,   callwild        1.5734155E-4 -4.736452618071205E-5
Also,   defoe-robinson-103.txt  1.645454E-5 -4.953310088506377E-6
```

Figure 3 Démonstration d'une partie des résultats finaux du programme TF-IDF

Q 5.2 Page Rank

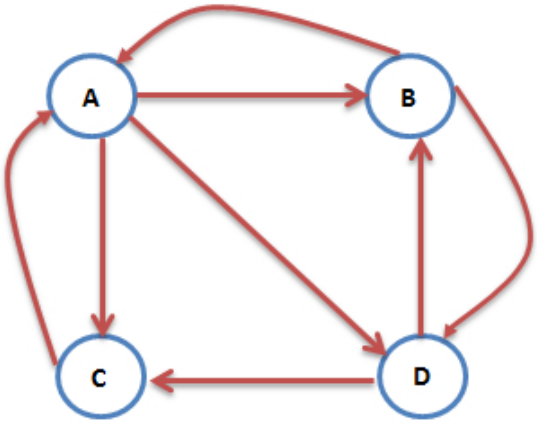
➢ PageRank

L'algorithme de PageRank est un algorithme de classement des pages web. Cet algorithme consiste à donner une valeur PR pour chaque page web. Intuitivement, on peut considérer la valeur PR comme la probabilité de visite du page web. Le PageRank est alors simplement la probabilité stationnaire d'une chaîne de Markov.

➢ Exemple de calcul

Exemple 1 :

Voici un simple graphe qui contient 4 pages
Le page A a pour antécédant B et C
Le page B a 2 successeurs (A et D)
Le page C a un seul successeur (A)

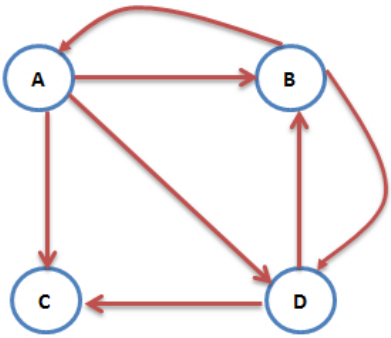


Exemple 1

Ainsi, $PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1}$

Exemple 2 :

Il existe des pages web qui n'ont pas de successeur, par exemple la page C.
On va considérer que la page C a 4 successeurs (y compris lui-même)
Le page A a pour antécédant B et C
Le page B a 2 successeurs (A et D)
Le page C a 4 successeur (A,B,C et D)



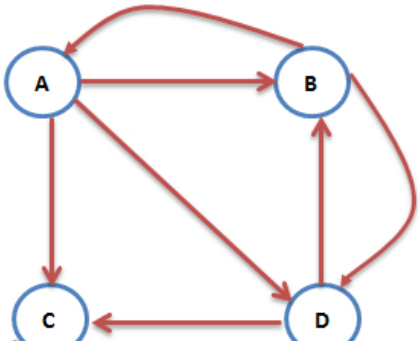
Exemple 2

Ainsi, $PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{4}$

Exemple 3 :

Dans ce graph, C a seulement lui-même comme successeur
On considère que si quelqu'un visite la page C,
Il va visiter aléatoirement une page avec la probabilité α

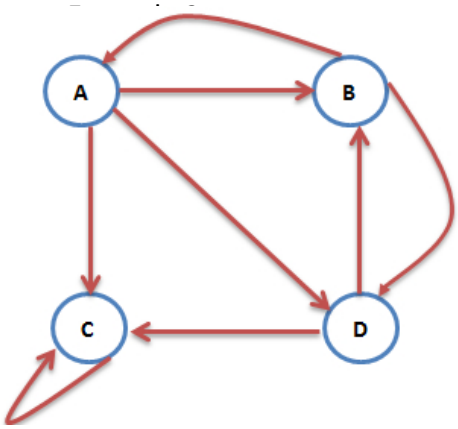
$$PR(A) = \alpha(\frac{PR(B)}{2}) + \frac{1-\alpha}{4}$$



Remarque : généralement on prend $\alpha = 0.85$

- Calcul des valeurs PR

On prend l'exemple 3



On définit une matrice S la matrice de probabilité

$$S = \begin{pmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 1 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{pmatrix}$$

Et on définit la matrice A

$$A = \alpha S + \frac{(1-\alpha)}{N} e e^T$$

Avec ces deux définitions, on peut calcule le vecteur PR en utilisant la relation suivante

$$P_{n+1} = A P_n$$

La limite de la suite Pn est alors le vecteur R (étant donné P0)

- Pourquoi utiliser MapReduce sur PageRank ?

Le vecteur PR converge généralement au bout d'une trentaine itérations. Si on a une matrice 4*4 comme ci-dessus, il n'y a aucun problème. Mais dans la vraie vie, il en a de milliards de pages web, d'où l'intérêt d'utiliser MapReduce.

- Etape 1 : Transforme la data

On transforme la data sous forme de **X A, B, C…** avec X la page web et A B C ses successeurs :

0	4,5,7,8,9...
4	1,10,12,...
5	0,1,2,4,8,...
.....	.....
75887	52098

- Etape 2 : Mapping

On va donner une clé pour chaque page web. Initialement, la clé est lui-même. Au bout de premier entrée, la clé est transformée en 'numéro de page web' + 'la valeur rank'. La structure de data après mapping est *key->{page1;rank1;count1,page2;rank2;count2,...,page1,page2...}*

Remarque : au début la valeur rank vaut 0.85 (c'est le α)

0,0.85	4,5,7,8,9...
4,0.85	1,10,12,...
5,0.85	0,1,2,4,8,...
.....	.....
75887,0.85	52098

- Etape 3 : Reducing

En considérant pour chaque clé la valeur de PageRank et le nombre de chaine sortant, on calcule la nouvelle valeur PageRank pour cette clé.
On remplace ensuite la clé par clé + ' , ' + valeur PageRank, et la valeur par [page1,page2,...pagen].

- Etape 4 : Itération

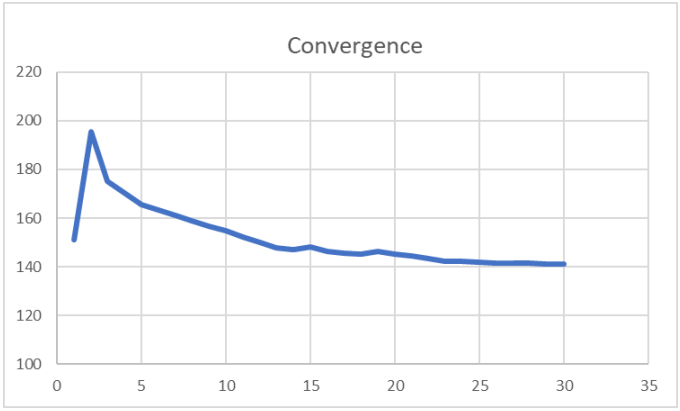
On va itérer 30 fois. Selon la théorie, au bout de 30 fois, la matrice est quasiment stationnaire.

- Etape 5 : Classement des pages

Les pages sont classés directement par leur score P ageRank

```
18    141.178
737   94.801
1719  67.073
118   66.0368
143   64.127
136   62.196
790   62.021
40    58.004
1619  47.140
725   46.905
```

On essaie de voir si la valeur PageRank converge, on prend par exemple la note de page 18



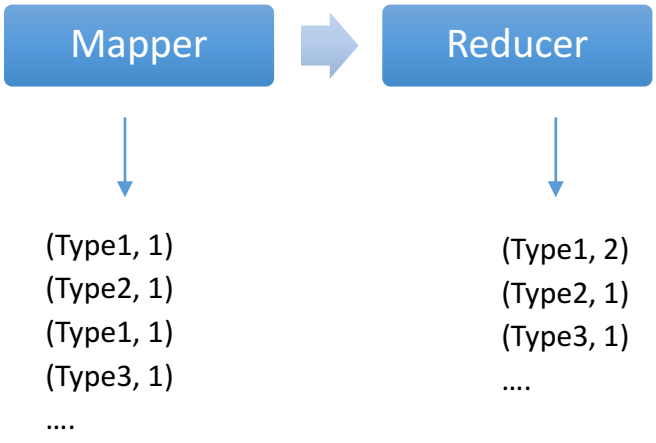
Question 5.3 The trees of Paris

L'objectif de cette question est d'écrire quelques programmes de MapReduce qui résoudent les questions suivantes:

- Déduire le nombre d'arbres par genre
- Déduire l'hauteur du plus haut arbre de chaque genre

Pour résoudre les questions, nous avons utilisé une structure simple qui contient Mapper et Reducer:

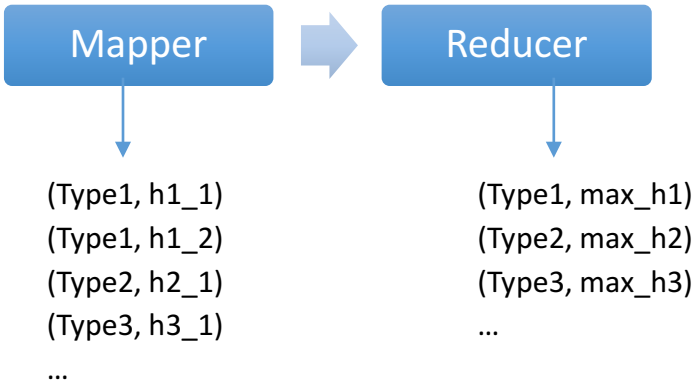
1) Déduire le nombre d'arbres par genre



Voici le résultat obtenu :

```
Acer    3
Aesculus        3
Ailanthus       1
Alnus   1
Araucaria       1
Broussonetia    1
Calocedrus      1
Catalpa 1
Cedrus  4
Celtis  1
Corylus 3
Davidia 1
```

2) Déduire l'hauteur du plus haut arbre de chaque genre



Voici le résultat obtenu:

```
Acer    16.0
Aesculus        30.0
Ailanthus       35.0
Alnus   16.0
Araucaria       9.0
Broussonetia    12.0
Calocedrus      20.0
Catalpa 15.0
Cedrus  30.0
Celtis  16.0
Corylus 20.0
Davidia 12.0
```

Dans cette question, nous avons 3 arguments : args[0], args[1], args[2]

Dans le args[0] c'est le [input], il faut mettre le fichier *arbre.csv.*
Pour le args[1], c'est [output1] pour la première petite question, et args[2], c'est [output2] pour la deuxième petite question.