Big Data Analytics TP1

Initiation à Hadoop, HDFS et MapReduce

Cécile Bothorel, Romain Picot-Clemente
Printemps 2015

1 Objectif

Cette séance de TP constitue un premier pas vers l'utilisation d'outils Big Data. Nous avons fait le choix de l'environnement Apache Hadoop, car c'est un framework open-source de stockage et de traitement de données volumineuses sur un cluster de machines distribuées. Il est utilisé par un grand nombre de contributeurs et utilisateurs. Il a une licence Apache 2.0.

Vous manipulerez des fichiers dans le système de fichiers distribué HDFS et vous lancerez des traitements MapReduce.

Au terme de ce TP, vous serez capables :

- comprendre les concepts de HDFS et MapReduce et comment on peut les utiliser pour résoudre des problèmes simples de comptage,
- de manipuler des fichiers dans le système de fichiers distribués HDFS,
- de développer une chaîne de traitement Hadoop/MapReduce (un Mapper et un Reducer) en Python,
- de tester en local sur votre propre machine si la chaîne de traitement est conforme à vos attentes,
- de déployer et exécuter votre chaîne de traitement MapReduce sur une machine virtuelle simulant un cluster Hadoop,
- ... et de rapatrier vos résultats d'analyse en local sur votre machine.

2 Installation

Ce TP est une adaptation d'un TP proposé par Institut National des Sciences Appliquées et de Technologie Tunisie, lui-même inspiré de la formation "*Intro to Hadoop and Map Reduce*" fait par Cloudera[1] et publié sur Udacity[2]. Cloudera fournit une machine virtuelle où Hadoop,

[1] Cloudera : Plateforme de BigData `https://www.cloudera.com/`

[2] Udacity : Plateforme de eLearning `https://www.udacity.com/` ; machine virtuelle `http://content.udacityŋdata.com/courses/ud617/Cloudera-Udacity-Training-VM-4.1.1.c.zi`

ainsi qu'un grand nombre d'outils de son écosystème, sont préinstallés. Nous utilisons cette machine virtuelle que nous avons fait installée sur les postes des salles de TP, elle s'appelle `VM cloudera-training2015`.

Démarrage :

- Démarrez votre machine virtuelle
- La machine virtuelle est en anglais par défaut, gênant pour le clavier... Pour le changer, allez dans la barre de menu de la distribution Linux : System –> Preferences –> Keyboard –> Layouts
- Lancez un terminal
- Sous le répertoire `~/udacity_training/data`, vous trouverez deux sous-répertoires : `code` et `data` dans lesquels on trouvera et on sauvegardera respectivement les codes de nos mappers et reducers, et les données sources et résultats.
- Déplacez-vous sous le répertoire `~/udacity_training/data`, et vérifiez que le fichier `purchases.txt` existe bien.

3 Manipulation de HDFS

Toutes les commandes interagissant avec le système de fichiers Hadoop commencent par `hadoop fs`. Ensuite, les options rajoutées sont très largement inspirées des commandes Unix standard.

Créez un répertoire dans HDFS, appelé `myinput`. Pour cela, tapez :

```
hadoop fs -mkdir myinput
```

Pour copier le fichier `purchases.txt` dans HDFS sous le répertoire `myinput`, il s'agit e se placer dans le répertoire local `data` où se trouve le fichier, puis tapez la commande :

```
hadoop fs -put purchases.txt myinput/
```

Pour afficher le contenu du répertoire `myinput`, la commande est :

```
hadoop fs -ls myinput
```

Pour **visualiser les dernières lignes du fichier**, tapez :

```
hadoop fs -tail myinput/purchase.txt
```

On obtient :

```
[training@localhost udacity_training]$ hadoop fs -ls myinput
Found 1 items
-rw-r--r--  1 training supergroup  211312924 2015-01-23 09:43 myinput/purchase.txt
[training@localhost udacity_training]$ hadoop fs -tail myinput/purchase.txt
31 17:59 Norfolk Toys 164.34 MasterCard
2012-12-31 17:59 Chula Vista Music 380.67 Visa
2012-12-31 17:59 Hialeah Toys 115.21 MasterCard
2012-12-31 17:59 Indianapolis Men's Clothing 158.28 MasterCard
2012-12-31 17:59 Norfolk Garden 414.09 MasterCard
2012-12-31 17:59 Baltimore DVDs 467.3 Visa
2012-12-31 17:59 Santa Ana Video Games 144.73 Visa
2012-12-31 17:59 Gilbert Consumer Electronics 354.66 Discover
2012-12-31 17:59 Memphis Sporting Goods 124.79 Amex
2012-12-31 17:59 Chicago Men's Clothing 386.54 MasterCard
2012-12-31 17:59 Birmingham CDs 118.04 Cash
2012-12-31 17:59 Las Vegas Health and Beauty 420.46 Amex
2012-12-31 17:59 Wichita Toys 383.9 Cash
2012-12-31 17:59 Tucson Pet Supplies 268.39 MasterCard
2012-12-31 17:59 Glendale Women's Clothing 68.05 Amex
2012-12-31 17:59 Albuquerque Toys 345.7 MasterCard
2012-12-31 17:59 Rochester DVDs 399.57 Amex
2012-12-31 17:59 Greensboro Baby 277.27 Discover
2012-12-31 17:59 Arlington Women's Clothing 134.95 MasterCard
2012-12-31 17:59 Corpus Christi DVDs 441.61 Discover
[training@localhost udacity_training]$
```

Dans la Table 1 nous résumons les commandes les plus utilisées dans Hadoop HDFS. La commande `hadoop version` donne Hadoop 2.0.0-cdh4.1.1, `hadoop fs` liste les commandes de manipulation du File System Shell. Une documentation est disponible ici, même si ce n'est pas l'exact version et qu'il existe des petites variantes : `http://archive.cloudera.com/cdh4/cdh/4/hadoop/hadoop-project-dist/hadoop-common/CommandsManual.html#fs`.

Activité 1
Testez les différentes fonctions citées ci-dessus pour :

- Créer sur HDFS un répertoire appelé `myinput`
- Copier le fichier local `purchases.txt` dans le répertoire `myinput`
- Afficher les dernières lignes du fichier distant

`hadoop fs -ls`	Afficher le contenu du répertoire racine
`hadoop fs -put file.txt`	Upload un fichier dans hadoop (à partir du répertoire courant linux)
`hadoop fs -get file.txt`	Download un fichier à partir de hadoop sur votre disque local
`hadoop fs -tail file.txt`	Lire les dernières lignes du fichier
`hadoop fs -cat file.txt`	Affiche tout le contenu du fichier
`hadoop fs -cat file.txt | less`	Lire le fichier page par page
`hadoop fs -mv file.txt newfile.txt`	Renommer le fichier
`hadoop fs -rm newfile.txt`	Supprimer le fichier
`hadoop fs -mkdir myinput`	Créer un répertoire
`hadoop fs -rm -f -r myinput`	Supprime un répertoire, et son contenu récursivement

TABLE 1 – Principales commandes de manipulation de fichiers HDFS

4 Map Reduce

Map Reduce est un patron d'architecture de développement permettant de traiter les données volumineuses de manière parallèle et distribuée.
Il se compose principalement de deux types de programmes : Les Mappers et les Reducers. Les Mappers permettent d'extraire les données nécessaires sous forme de clef/valeur, pour pouvoir ensuite les trier selon la clef. Les Reducers prennent un ensemble de données triées selon leur clef, et effectuent le traitement nécessaire sur ces données (somme, moyenne, total...).
Pour notre TP, nous utilisons le langage Python pour développer les Mappers et les Reducers. Les traitements intermédiaires (comme le tri par exemple) sont effectués automatiquement par Hadoop.

4.1 Mapper

Soit un fichier comportant 6 champs, séparés par des tabulations. Le Mapper doit :

- Séparer les différents champs par tabulation
- Extraire les éléments voulus à partir de ces champs, sous forme de clef/valeur

Pour ce premier exercice, notre but est de déterminer le total des ventes par magasin, pour un fichier log dont les champs sont de la forme suivante :
`date -> temps -> magasin -> produit -> coût -> paiement`
Pour calculer les ventes par magasin, le couple (clef, valeur) à extraire est `(magasin,coût)`.
Pour faire cela, le code du Mapper est le suivant :

```
#!/usr/bin/python

# Format of each line is:
# date\ttime\tstore name\titem description\tcost\tmethod of payment
#
# We want elements 2 (store name) and 4 (cost)
# We need to write them out to standard output, separated by a tab

import sys
```

```
for line in sys.stdin:
    data = line.strip().split("\t")
    if len(data) == 6:
        date, time, store, item, cost, payment = data
        print "{0}\t{1}".format(store, cost)
```

Ce code se trouve sous le répertoire `/udacity_training/code` dans le fichier `mapper.py`[3].

Activité 2

- Que permet de faire chaque ligne de ce code ?
- Testez ce Mapper en local sur les 50 premières lignes du fichier `purchases.txt` en tapant l'instruction suivante, directement à partir de votre répertoire `code` :

```
head -50 ../data/purchases.txt | ./mapper.py
```

4.2 Reducer

Le Reducer permet de faire le traitement désiré sur des entrées sous forme de clef/valeur, préalablement triées par Hadoop (on n'a pas à s'occuper du tri manuellement). Dans l'exemple précédent, une fois que le Mapper extrait les couples (store,cost), le Reducer aura comme tâche de faire la somme de tous les coûts pour un même magasin. Le code du Reducer est le suivant :

```
#!/usr/bin/python

# Format of each line is:
# date\ttime\tstore name\titem description\tcost\tmethod of payment
#
# We want elements 2 (store name) and 4 (cost)
# We need to write them out to standard output, separated by a tab

import sys

salesTotal = 0
oldKey = None

# Loop around the data
# It will be in the format key\tval
# Where key is the store name, val is the sale amount
```

[3] Remarque : Python est un langage qui délimite les différents blocs en utilisant les tabulations, faites alors bien attention à vos indentations !

```
#
# All the sales for a particular store will be presented,
# then the key will change and we'll be dealing with the next store

for line in sys.stdin:
    data_mapped = line.strip().split("\t")
    if len(data_mapped) != 2:
        # Something has gone wrong. Skip this line.
        continue

    thisKey, thisSale = data_mapped

    if oldKey and oldKey != thisKey:
        print oldKey, "\t", salesTotal
        oldKey = thisKey;
        salesTotal = 0

    oldKey = thisKey
    salesTotal += float(thisSale)

if oldKey != None:
    print oldKey, "\t", salesTotal
```

Ce code se trouve sous le répertoire `/udacity_training/code` dans le fichier `reducer.py`.

Activité 3

- Que permet de faire chaque ligne de ce code ?
- Testez ce Reducer en local sur les 50 premières lignes du fichier `purchases.txt` en tapant l'instruction suivante, directement à partir de votre répertoire `code` :

```
head -50 ../data/purchases.txt | ./mapper.py | sort |./reducer.py
```

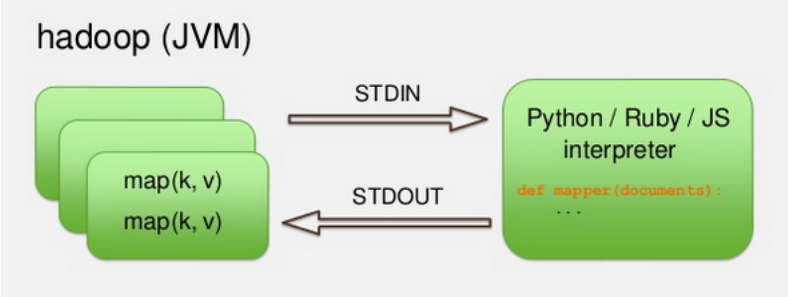
4.3 Lancer un Job entier

Lancer un job entier sur Hadoop implique qu'on fera appel au mapper puis au reducer sur une entrée volumineuse, et qu'on obtiendra à la fin un résultat, directement sur HDFS. Pour faire cela, l'instruction à exécuter est :

```
hadoop jar /usr/lib/hadoop-0.20-mapreduce/contrib/streaming/hadoop-streaming-
2.0.0-mr1-cdh4.1.1.jar -mapper mapper.py -reducer reducer.py -file mapper.py
-file reducer.py -input myinput -output joboutput
```

Cette instruction donne en paramètres les fichiers correspondant aux Mappers et Reducers, et les répertoires contenant le fichier d'entrée (`myinput`) et la sortie à générer (`joboutput`). Le répertoire de sortie, après exécution, contiendra un fichier appelé `part-00000`, représentant la sortie désirée.

Remarque 1 : Nous utilisons `Hadoop Streaming` qui permet de créer et lancer des jobs MapReduce avec tout type d'exécutable ou script en tant que mapper et reducer. La manière standard est d'écrire des programmes MapReduce en Java via l'API Java MapReduce. Ici nos scripts sont écrits en Python, mais les mappers et reducers pourraient être des classes Java, des utilitaires unix, des scripts R, Ruby, etc. Les Mappers liront les données fournies dans le flux standard d'entrée unix `stdin` et les réécriront dans la sortie standard `stdout` via `print`.



Remarque 2 : Le répertoire d'entrée doit contenir un seul fichier. Le répertoire de sortie ne doit pas exister avant l'exécution de l'instruction.
Pour faciliter le travail, un raccourci a été créé pour cette instruction (vous le trouverez défini dans le fichier `~/.bashrc`). Désormais, pour exécuter cette instruction, il suffit de taper :

```
hs mapper.py reducer.py myinput joboutput
```

Activité 4

- Exécutez un job hadoop sur le fichier `purchases.txt` en utilisant les fichiers `mapper.py` et `reducer.py` déjà fournis. Stockez le résultat dans un répertoire `joboutput`. Sauvegardez ensuite le fichier `part-00000` dans votre répertoire local.
- Quelle est la totalité des ventes du magasin de Buffalo ?

5 A vous de jouer !

Nous continuons à travailler avec le même fichier en entrée (`purchases.txt`), mais pour obtenir des résultats différents. Le but est donc d'écrire vos propres Mappers et Reducers.

Activité 5

- Donnez la liste des ventes par catégorie de produits.
- Quelle est la valeur des ventes pour la catégorie Toys ?
- Et pour la catégorie Consumer Electronics ?

Activité 6

- Donnez le montant de la vente le plus élevé pour chaque magasin
- Quelle est cette valeur pour les magasins suivants : Reno ? Toledo ? Chandler ?
- Quelle est la valeur des ventes pour la catégorie Toys ?

Activité 7

- Quel est le nombre total des ventes et la valeur totale des ventes de tous magasins confondus ?

Soit un fichier Log qui stocke les accès aux différentes pages d'un site web donné. Ce fichier se trouve dans le répertoire `~/udacity_training/data` et s'appelle `access_log`. Une entrée de ce fichier est composée de plusieurs champs séparés par des espaces. Elle a la forme suivante :

```
10.223.157.186 - - [15/Jul/2009:15:50:35 -0700] "GET /assets/js/lowpro.js
HTTP/1.1" 200 10469

%h %l %u %t "%r" %s %b

Où :

   • %h est l'adresse IP du client
   • %l est l'identité du client (- si indisponible)
   • %u est le nom d'utilisateur du client (- si indisponible)
   • %t est la date de fin de traitement de la requête par le serveur. Le format
     est [jour/mois/année:heure:minute:seconde zone]
   • %r est la requête du client (donnée entre guillemets). Elle contient la
     méthode, le chemin du fichier, la requête et le protocole.
```

- `%s est le code du statut que le serveur envoie au client : 200 (OK, la requête a été exécutée avec succès), 304 (Non modifiée) et 404 (Introuvable).`
- `%b la taille de l'objet retourné au client en octets (- si le statut est 304).`

Activité 8

- Ecrire un code MapReduce permettant de donner, pour chaque page du site web, le nombre de fois où elle a été accédée. Combien de fois est-ce que la page `/assets/js/the-associates.js` a-t-elle été accédée ?
- Ecrire un code MapReduce permettant de donner le nombre d'accès au site par chaque adresse IP. Combien de fois l'adresse 10.99.99.186 a-t-elle accédé au site ?
- Trouver le fichier le plus populaire du site web (le fichier qui a été accédé le plus de fois). Le Reducer doit juste afficher le nom du fichier et le nombre d'accès dans le fichier résultat. Quel est le nom de ce fichier ? Le nombre d'accès ?