Telecom Nancy
2A, IAMD et SIE
Adrien Coulet et Mario Lezoche
Date de dernière mise à jour : 28/11/2013

TP encadré n°5 :

"Exécuter un processus MapReduce avec Hadoop"

OBJECTIF

Hadoop permet de distribuer un traitement sur de grands volumes de données en utilisant notamment le système de fichier distribué HDFS et le paradigme de programmation MapReduce.
Dans ce TP vous devez comprendre, exécuter et modifier un processus MapReduce. Dans un premier temps vous testerez vos programmes en local, puis les lancerez sur un cluster.

SOFTWARE

Téléchargez la version 1.2.1 de la librairie Hadoop à partir du miroir du projet apache : http://www.apache.org/dyn/closer.cgi/hadoop/common/ (le fichier hadoop-1.2.1-bin.tar.gz fera l'affaire).
Dans votre projet Java, ajoutez au classpath tous les .jar la librairie Hadoop qui sont à la racine et dans le dossier lib/

DOCUMENTATION

Tutorial MapReduce : http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

QUESTIONS

Exercice 1 : Comprendre et tester en local

L'objectif de ce premier exercice est de comprendre et d'exécuter le code Java **WordCount** qui instancie un processus MapReduce pour compter le nombre d'occurrences des mots des plusieurs textes. C'est un exemple classique qui illustre MapReduce (cf. Cours 4 de GMD).
Vous trouverez le code à comprendre puis à exécuter à l'url suivante :
http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html#Example%3A+WordCount+v1.0
Pour tester votre code vous pourrez utiliser, en entrée, le texte du fichier suivant :
http://www.loria.fr/~coulet/teaching/gmd/data.txt
Choses importantes à comprendre avant de passer à la suite :

- -instanciation des fonctions Map et Reduce ;
- -gestion des entrées et sortie ;
- -création, configuration et paramétrage d'un job.

a) Exécutez le programme en local d'abord à partir de votre IDE (Integrated Development Environment)
b) Puis, exécutez le programme en local à partir d'une ligne de commande de la forme suivante :

```
hadoop --config ./fichierDeConfig ClassAExecuter ./fichierEnEntree
DossierDeSortie
```

où l'argument `--config` permet de passer en paramètre un fichier de configuration pour l'execution local de hadoop. Vous pouvez utiliser le fichier suivant :
http://www.loria.fr/~coulet/teaching/gmd/hadoop-local.xml

Exercice 2 : Packager puis tester sur un cluster

a) Faites un .jar de votre projet en prenant soin d'intégrer au jar les dépendances et testez le en local avec une commande de la forme suivante :

```
hadoop --config ./fichierDeConfig jar mon_jar.jar ./fichierEnEntree
DossierDeSortie
```

b) Copiez votre jar dans un sous dossier de votre répertoire personnel sur le « nœud maître » du cluster **hadoop-master.telecomnancy.univ-lorraine.fr**

c) Executez votre jar avec hadoop sur le cluster avec la même commande que en a) en prenant soins de remplacer le fichier de configuration pour une exécution locale avec le fichier de configuration pour une exécution sur le cluster. Vous pouvez utilisez le fichier suivant :
http://www.loria.fr/~coulet/teaching/gmd/hadoop-cluster.xml

d) Regardez et comprenez le contenu des deux fichiers de configuration

Exercice 3 : Modifier en local, exécuter sur le cluster

a) En local, modifiez et testez le programme WordCount pour compter le nombre d'occurrences de chaque mot présent dans la base de données DrugBank utilisée lors du TP4. Vous pouvez accéder à DrugBank
ici/home/depot/2A/gmd/tp4/drug_bank/drugbank.txt ou là
Sur le cluster **hadoop-master.telecomnancy.univ-lorraine.fr** :/users/data/drugbank.txt

b) Comptez (avec la fonction getTime() de la classe Java java.util.Date) le temps d'exécution en local.

c) Exécutez le même programme sur le cluster et comparez son temps d'exécution avec celui de l'exécution en local.

Exercice 4 :

a) Modifiez en local les fonctions Map et Reduce de sorte à ne considérer que de vrais mots lors du comptage. L'objectif est d'avoir en sortie du traitement de DrugBank un ensemble de mots nettoyés, plus facile à interpréter. Ainsi :

- si le mot commence par un caractère non alphabétique, comme un guillemet, retirez le ;
- si le mot n'est pas un mot mais par exemple un nombre ou une séquence biologique (d'ADN ou de protéine comme ACTGACTG ou MNTRSMWNTRS) ne le comptez pas,
- etc.

b) Testez sur le cluster.

Exercice 5 :

a) Modifiez en local le code pour que les résultats écrits dans le fichier `part-00000` soient triés par ordre de fréquence des mots.

b) Testez sur le cluster.