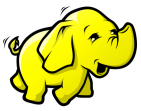
Travaux pratiques préparatoires : Hadoop

Jonathan Lejeune



Objectif

Ce sujet de travaux pratiques préparatoires est à faire au préalable de la séance correspondante. Il vous guidera dans l'installation d'un environnement de travail pour la plate-forme Hadoop sur UNIX.

Pré-requis

Le déploiement de Hadoop sur les machines du cluster (dans notre cas, il se limite à la machine locale) nécessite de se connecter via ssh pour chaque démon. Vous devez donc avoir configuré SSH pour pouvoir vous connecter en local sans mot de passe.

Introduction

Hadoop est une plate-forme distribuée écrite en Java et est composée de 2 services notables :

- **HDFS** (**H**adoop **D**istributed **F**ile **S**ystem) : est un système de fichiers distribué adapté aux gros volumes de données
- **YARN** (**Y**et **A**nother **R**esource **N**egotiator) : est un système de gestion de ressources de cluster à travers des conteneurs. C'est dans ces conteneurs que les tâches d'une application Yarn (ex : Hadoop MapReduce, Spark, ...) seront exécutées.

Bien que HDFS et YARN soient les composants principaux de la plate-forme Hadoop, ils peuvent être utilisés de manière complètement indépendante. Ces deux services sont ainsi autonomes l'un de l'autre et ont donc des fichiers de configuration et de scripts de déploiement qui leur sont propres. Ils fonctionnent tous les deux selon un schéma maître/esclave c'est-à-dire qu'ils déploient chacun sur les machines du cluster un processus serveur maître qui contrôle un ensemble de processus serveurs esclaves. Pour HDFS le serveur maître est appelé **NameNode** et les serveurs esclaves sont appelés **DataNode**. Pour YARN, le serveur maître est appelé **ResourceManager** et les serveurs esclaves sont appelés **NodeManager**.

On note également que Hadoop est fournie avec une application Yarn : Hadoop MapReduce. Cette application permet d'exécuter des calculs type map-reduce sur le cluster géré par Yarn et HDFS. Cette application implante le rôle métier historique de Hadoop (version 1) qui autrefois n'était dédié qu'au calcul MapReduce sur un cluster de machine. L'arrivée de la version 2 a permis de séparer la gestion des ressources de calcul (Yarn) et la partie calcul map-reduce (Hadoop MapReduce). La plate-forme Hadoop peut maintenant être considérée comme un système d'exploitation réparti adapté au stockage de données massives ainsi qu'au déploiement d'application de leur traitement .

Hadoop dispose de trois modes de fonctionnement :

- **le mode local** : un seul processus pour simuler la plate-forme entière ce qui ne nécessite pas d'avoir des connexions ssh. Ce mode s'utilise principalement pour développer et debuguer des programmes Map-Reduce avant de les exécuter en production. Le système de fichier n'est donc pas le HDFS mais le système de fichiers local.
- **le mode pseudo-distribué** : une instance de chaque démon de Hadoop est lancée sur la machine locale. Avec le loopback, il est ainsi possible d'émuler un réseau de machines distantes. Cependant il n'est pas possible de lancer plus d'une instance de chaque démon à cause des conflits de port d'écoute.
- **le mode distribué** : Hadoop est déployé et exécuté sur un système distribué réel (cluster, grille de calcul, cloud). C'est le mode que l'on utilise en production.

Dans notre cadre, nous fonctionnerons en mode pseudo-distribué afin de prendre en compte les différentes contraintes liées à la communication entre les démons. Il vous ai cependant possible (en lisant la documentation en ligne) de tester vos programmes :

- en mode local si jamais vous rencontrez des problèmes de configuration ou de connexion
- en mode distribué sur plusieurs machines. Dans ce cas, prenez vos précautions en respectant la chartre d'utilisation des machines.

Exercice 1 – Installation et démarrage de la plate-forme Hadoop en mode pseudo-distribué

Étape 1

Téléchargez la dernière version stable de la plateforme Hadoop à l'URL suivante.

`http://miroir.univ-lorraine.fr/apache/hadoop/common/stable/`

Extrayez le contenu de hadoop-2.x.x.tar.gz dans votre home. Pour fonctionner, Hadoop a besoin que la variable d'environnement `JAVA_HOME` soit définie pour indiquer le répertoire d'une jvm. Vous pouvez connaître ce chemin avec la commande :

```
echo $(dirname $(dirname $(readlink -f $(which javac))))
```

Ajoutez ensuite les lignes suivantes dans votre *.bashrc* :

```
export JAVA_HOME=<chemin vers la JVM courante de votre machine>
export MY_HADOOP_HOME=<votre_repertoire_hadoop>
export PATH=$PATH:$MY_HADOOP_HOME/bin
export PATH=$PATH:$MY_HADOOP_HOME/sbin
export HADOOP_HOME=$MY_HADOOP_HOME
```

Pour prendre en compte ces modifications dans votre terminal ouvert tapez :

```
source ~/.bashrc
```

Vérifiez que Hadoop est opérationnel en tapant :

```
hadoop version
```

La sortie terminale de cette commande doit avoir le format suivant :

```
Hadoop 3.X.Y
Source code repository ..................................
Compiled by .............................................
Compiled with ...........................................
From source with checksum ...............................
This command was run using ..............................
```

Étape 2

Les différents fichiers exécutables fournis par Apache sont stockés dans les répertoires *bin* et *sbin* de votre répertoire Hadoop. Ces exécutables sont en grande majorité des scripts shells. Les fichiers stockés dans le répertoire *bin* sont des commandes génériques qui permettent d'administrer, de configurer la plateforme, d'utiliser le hdfs ou d'exécuter des applications Yarn (ex : programme Map Reduce). Parmi ces fichiers nous pouvons citer principalement la commande :

- *hdfs* : permet de manipuler le système de fichiers distribué HDFS
- *yarn* : permet de manipuler le gestionnaire de ressources YARN
- *mapred* : permet de manipuler/administrer des jobs de Hadoop Map Reduce
- *hadoop* : ancienne commande qui réunissait les trois précédentes dans la version 1.x d'Hadoop. Cette commande est dépréciée à ce jour et sert principalement à afficher la version de la plateforme.

Pour en faciliter l'utilisation, un certain nombre de scripts sont livrés avec elle dans le répertoire *sbin*. Ces scripts encapsulent les différents appels aux scripts ci-dessus permettant de réaliser des actions complexes notamment pour déployer la plateforme sur le cluster. En pratique, vous aurez besoin des scripts : *start-dfs.sh*, *stop-dfs.sh*, *start-yarn.sh* et *stop-yarn.sh*.

Vous pouvez éventuellement regarder le contenu de ces scripts pour comprendre le déploiement de la plate-forme

Étape 3

Avant de lancer la plate-forme Hadoop, il faut éditer ses fichiers de configuration se trouvant dans le dossier *etc/hadoop*. Dans ce TP nous nous limiterons à l'utilisation d'Hadoop en mode pseudo-distribué, c'est à dire lancer tous les démons Hadoop sur une seule machine (la machine locale). Vous devez donc configurer Hadoop de la façon suivante :

- le fichier *etc/hadoop/workers* (utilisé pour automatiser les connections ssh) permet d'indiquer le nom de la ou des machines utilisées, ici cela sera votre machine locale désignée par :

  ```
  localhost
  ```

- le fichier *etc/hadoop/hdfs-site.xml* est le fichier de configuration du HDFS. Dans le cadre de ce TP son contenu doit être (Attention : les copier-coller depuis le

pdf pouvant engendrer des caractères cachés, il est conseillé de taper à la main le contenu) :

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

Ceci indique au HDFS que les blocs ne doivent pas être répliqués (au lieu de 3 réplications par défaut).

- le fichier *etc/hadoop/core-site.xml* est le fichier général de configuration de la plate-forme . Dans le cadre de ce TP son contenu doit être :

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

Ceci indique l'URL du processus serveur Namenode du HDFS. Vérifiez au préalable que le port 9000 de votre machine n'est pas déjà utilisé (commande "`netstat -a | grep tcp | grep LISTEN | grep 9000`" qui ne doit rien afficher).

- le fichier *etc/hadoop/mapred-site.xml* (à créer si ce dernier n'existe pas) est le fichier permettant de configurer l'application Hadoop MapReduce permettant d'exécuter des calculs map-reduce. Dans le cadre de ce TP son contenu doit être :

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
        <name>mapreduce.admin.user.env</name>
        <value>HADOOP_MAPRED_HOME=$HADOOP_COMMON_HOME</value>
    </property>
    <property>
        <name>yarn.app.mapreduce.am.env</name>
        <value>HADOOP_MAPRED_HOME=$HADOOP_COMMON_HOME</value>
    </property>
</configuration>
```

La première property indique que l'on utilise le mode distribué du map reduce en

utilisant la plateforme YARN. Les deux autres permettent une compatibilité des variables d'environnement entre Hadoop 2 et Hadoop 3.

- le fichier *etc/hadoop/yarn-site.xml* permettant de configurer YARN. Dans le cadre de ce TME son contenu doit être :

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

Attention : avant de passer à la suite, vérifiez bien que chaque fichier XML comporte bien une seule et unique balise <?xml .. ?> et <?xml−stylesheet .. ?>

Étape 4

Nous allons maintenant démarrer la plate-forme, c'est-à-dire lancer l'ensemble des démons d'Hadoop :

1. Formatez le HDFS :

   ```
   hdfs namenode -format
   ```

2. Démarrez le HDFS puis Yarn :

   ```
   start-dfs.sh
   start-yarn.sh
   ```

3. Attendez quelques secondes une fois que ces commandes sont terminées et vérifiez le bon démarrage d'Hadoop avec le script *check_ start.sh* qui vous ai fourni dans les ressources de TP.

   ```
   check_start.sh hdfs
   check_start.sh yarn
   ```

Une fois la plate-forme lancée, utilisez la commande *ps uxf* pour comprendre quels démons ont été lancés. Vous pourrez constater que 5 processus java ont été lancés.

Étape 5

Une fois que la plate-forme est démarrée, chaque démon démarre un serveur web sur la machine hôte afin de pouvoir superviser son état à l'aide d'un navigateur. La commande

```
netstat -na | grep tcp | grep LISTEN
```

vous liste les différents serveurs à l'écoute sur votre machine avec leur numéro de port associé. Avec votre navigateur, trouvez à quel service web chaque serveur correspond (ne s'intéresser qu'aux numéros de port supérieurs à 1000).

Exercice 2 – Préparation de Eclipse

Afin d'utiliser *Eclipse* pour éditer et compiler des programmes pour l'application Hadoop Map Reduce écrit en JAVA, il faut déclarer une nouvelle librairie permettant de déclarer dans le build path d'un projet Eclipse (= le classpath du point de vue de la JVM) les différents fichiers jars de Hadoop.

Étape 1

Pour ce faire :

- Dans la barre de menu de Eclipse cliquez sur *Window*
- Cliquez sur *Preferences*
- Déroulez l'onglet *Java* sur le panneau de gauche
- Déroulez l'onglet *Build Path*
- Cliquez sur le menu *User Libraries*
- Cliquez sur le bouton *New...*
- Nommez la librairie `hadoop-3.x.y` (où `x` et `y` sont a remplacer par la version que vous avez téléchargée)
- Sélectionnez la nouvelle librairie créée et cliquez sur le bouton *Add External JARs...*
- Ajoutez l'ensemble des *Jar* présents à la racine des dossiers suivants : *share/hadoop/common*, *share/hadoop/common/lib*, *share/hadoop/hdfs*, *share/hadoop/mapreduce*, *share/hadoop/yarn* et *share/hadoop/tools/lib*
- cliquez sur *Apply and Close* pour valider.

Étape 2

Créez un projet java nommé *exercices_hadoop*.

Étape 3

Ajouter la librairie créée précédemment afin de programmer avec l'API Java de Hadoop. Pour ce faire :

- Sélectionnez *Build Path* dans le menu contextuel (clic droit sur le dossier) du projet puis *Add Libraries...*
- Sélectionnez *User Library* puis *Next >*
- Cochez la librairie puis cliquez sur *Finish*.
- Vérifiez que la librairie a bien été ajoutée au projet dans le Package Explorer d'Eclipse.

Étape 4

Copiez dans le dossier des sources de votre projet Eclipse le dossier *mapreduce* qui vous a été fourni. En intégrant ce dossier dans Eclipse, s'assurer que la compilation se passe bien (absence de croix rouge). Pour l'instant, ne vous préoccupez pas du code en lui-même, celui-ci sera analysé ultérieurement.

Exercice 3 – Test d'exécution

Nous allons maintenant nous assurer que l'exécution de job map-reduce se déroule bien en testant le programme *Wordcount* sur le fichier *loremIpsum* présent dans les ressources du TME.

Étape 1

Copiez le fichier *loremIpsum* sur le HDFS. Pour cela deux commandes équivalentes existent :

```
hdfs dfs -copyFromLocal <nomFichierLocal> /
```

ou bien

```
hdfs dfs -put <nomFichierLocal> /
```

La racine de la dernière commande indique la racine du HDFS et non pas la racine du système de fichier de votre machine.

Étape 2

Vérifiez que votre fichier est bien présent en tapant :

```
hdfs dfs -ls /
```

Étape 3

Créez à partir des fichiers compilés par Eclipse, une archive java *WordCount.jar*.

```
jar cvf <fichier jar a creer> -C <dossier de reference> <dossier a archiver>
```

Le dossier de référence correspond au dossier *bin* de votre projet Eclipse. Le dossier à archiver correspond au dossier *bin* lui même, sa valeur est donc égale à `"."` .

Étape 4

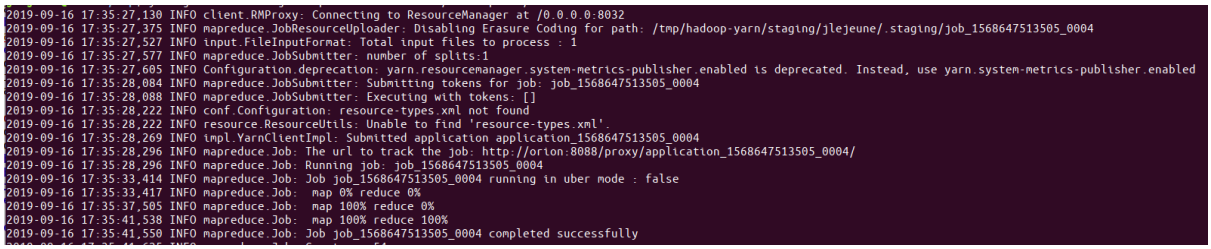
Lancez le WordCount avec le fichier jar produit précédemment avec la commande :

```
yarn jar <Fich jar> <nom absolu classe main> <fich entree> <dossier sortie>
```

Notons que les chemins d'entrée et de sortie sont des chemins HDFS et non des chemins du système de fichier local. Pour rappel, le nom absolu d'une classe est égal au nom de la classe préfixé par son package java.

Étape 5

Lors de l'exécution vous devez voir la progression en pourcentage des phases map et reduce du job. La sortie terminal doit ressembler à la figure ci-dessous :



Étape 6

Vous pouvez voir le résultat de votre calcul dans le dossier de sortie (HDFS) que vous avez spécifié dans votre commande.

- Pour lister les fichiers dans un chemin hdfs, tapez :
  `hdfs dfs -ls <chemin du dossier a lister>`
- pour afficher le contenu d'un fichier du hdfs, tapez :
  `hdfs dfs -cat <chemin du fichier a afficher>`